Automatica 119 (2020) 109015

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A reliability-aware multi-armed bandit approach to learn and select users in demand response☆

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

[a] Harvard University, Cambridge, MA 02138, USA
[b] Southeast University, Nanjing, China

ARTICLE INFO

Article history:
Received 15 May 2019
Received in revised form 23 March 2020
Accepted 8 April 2020
Available online 18 June 2020

Keywords:
Learning theory
Optimization under uncertainties
Real time simulation and dispatching
Multi-armed bandit
Demand response
Regret analysis

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 users, 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. Finally, we numerically test our algorithms using synthetic and real data, and demonstrate that our CUCB-Avg performs significantly better than the classic CUCB and also better than Thompson Sampling.

1. Introduction

Unknown and uncertain user behavior is common in many sequential decision-making problems of societal systems, such as transportation, electricity grids, communication, crowd-sourcing, and resource allocation problems in general (Belleflamme, Lambert, & Schwienbacher, 2014; Kuderer, Gulati, & Burgard, 2015; Li & Li, 2017; O'Neill, Levorato, Goldsmith, & Mitra, 2010). One key challenge caused by the unknown and uncertain user behavior is how to ensure reliability or reduce risks for the system. This paper focuses on addressing this challenge for residential demand response (DR) in power systems.

Residential DR refers to adjusting power consumption of residential users, e.g. by changing the temperature setpoints of air conditioners, to relieve the supply–demand imbalances of the power system (Edison, 2019; FERC, 2017; O'Neill et al., 2010; PSEG, 2019; ThinkEco, 2019). In most residential DR programs, customers can decide to respond to a DR signal or not, and the decisions are usually highly uncertain. Moreover, the pattern of the user behavior is not well understood by the DR aggregator. Such unknown and uncertain behavior may cause severe troubles for the system reliability: without enough knowledge of the user behavior, the DR load adjustment is likely to be very different from a target level, resulting in extra power imbalances and fluctuations. Therefore, it is critical for residential DR programs to learn the user behavior and ensure reliability during the learning.

Multi-armed bandit (MAB) emerges as a natural framework to learn the user behavior (Auer, Cesa-Bianchi, & Fischer, 2002; Bubeck, Cesa-Bianchi, et al., 2012). In a simple setting, MAB considers n independent arms, each providing a random contribution according to its own distribution at time step $1 \le t \le T$. Without knowing these distributions, a decision maker picks one arm at each time step and tries to maximize the total expected contribution in T time steps. When the decision maker can select multiple arms at each time, the problem is often referred to as combinatorial multi-armed bandit (CMAB) in literature (Chen, Wang, Yuan, & Wang, 2016; Kveton, Wen, Ashkan, & Szepesvari, 2015). (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 when the probability model is known and the expected value achieved by the online learning algorithm. It is desirable to design online algorithms with sublinear $o(T)$ regrets, which

☆ The work was supported by NSF, USA CAREER 1553407, NSF, USA ECCS 1839632, AFOSR, USA YIP, ONR, USA YIP, and ARPA-E, USA through the NODES program. The material in this paper was partially presented at The 57th IEEE Conference on Decision and Control, December 17-19, 2018, Miami Beach, Florida, USA. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: yingyingli@g.harvard.edu (Y. Li), qhu@seu.edu.cn (Q. Hu), nali@seas.harvard.edu (N. Li).

https://doi.org/10.1016/j.automatica.2020.109015
0005-1098/

roughly indicates that the learning algorithm eventually learns the optimal solution.

Though there have been studies on DR via (C)MAB, most literature aims at maximizing the load reduction (Jain, Narayanaswamy, & Narahari, 2014; Lesage-Landry & Taylor, 2017; Wang, Liu, & Mathieu, 2014). There is a lack of efforts on improving the reliability of CMAB algorithms for DR as well as the theoretical reliability guarantees.

1.1. Our contributions

In this paper, we formulate the DR as a CMAB problem with a reliability objective, i.e. we aim to minimize the deviation between the actual total load adjustment and a target signal. The target might be caused by a sudden change of renewable energy or a peak load reduction event. We consider a large number of residential users, and each user can commit one unit of load change with an unknown probability. The task of the DR aggregator is to select a subset of the users to guarantee the actual load adjustment to be as close to the target as possible. The number of users to select is not fixed, providing flexibility for the aggregator for achieving different target levels.

In order to design our online learning algorithm, we first develop an offline combinatorial optimization algorithm that selects the optimal subset of the users when the user behavior models 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 and $O(\sqrt{T\log(T)})$ regret given time-varying targets. The regrets in both cases depend polynomially on the number of users n. We also conduct numerical studies using real DR data and show that the performance of CUCB-Avg is much better than the classic algorithm CUCB (Chen et al., 2016; Kveton et al., 2015), and also better than Thompson sampling (Russo, Van Roy, Kazerouni, & Osband, 2017). In addition, we numerically show that, with minor modifications, CUCB-Avg can cope with more realistic behavior models with user fatigue effects.

Lastly, we would like to mention that though the DR model considered in this paper is very simple, the model is motivated by real pilot studies of residential DR programs, and the results have served as a guideline for designing the learning protocols in DR programs (ThinkEco, 2019). In addition, since real-world DR programs vary a lot among each other (depending on the DR companies, local policies, reward schemes, data infrastructure, etc.), abstracting the DR model can be useful for a broad range of DR programs by providing common insights and general guidelines. When designing algorithms for different DR programs, we could modify the vanilla method to suit specific requirements. Further, this paper may also provide insights for other societal system applications.

1.2. Related work

Combinatorial multi-armed bandits. There is a rich body of literature in CMAB aiming to maximize the total (weighted) contribution of K arms with a *fixed* integer K (and known weights) (Bubeck et al., 2012; Kveton et al., 2015). There are also papers considering more general reward functions, for example, Chen et al. (2016) consider objective functions that are *monotonically nondecreasing* with the parameters of the selected arms and design Combinatorial Upper Confidence Bound (CUCB) under the principle of *optimism in the face of uncertainty*. However, the reliability objective of our CMAB problem does not satisfy the monotonicity assumption, thus the study of CUCB cannot be directly applied here. Another line of work follows the *Bayesian* approach and studies Thompson sampling (Gopalan, Mannor, & Mansour, 2014; Wang & Chen, 2018). However, the regret bound of Thompson sampling consists of a term that is independent of T but depends exponentially on the number of arms K^* in the optimal subset (Wang & Chen, 2018). Further, Wang and Chen (2018) show that the exponential dependence is unavoidable. In the residential DR problems, K^* is usually large, so Thompson sampling may generate poor performance especially when T is not very large, which is consistent with our numerical results in Fig. 3 in Lemma 5. Finally, there is a lack of analysis on time-varying objective functions, but in many real-world applications the objectives change with time, e.g., the DR target would depend on the time-varying renewable generation. Therefore, either the learning algorithms or the theoretical analysis in literature does not directly apply to our CMAB problem, which motivates this paper.

Risk-aversion MAB. There is a related line of research on reducing risks in MAB by selecting the *single* arm with the best *return–risk tradeoff* (Sani, Lazaric, & Munos, 2012; Vakili & Zhao, 2016). However, there is a lack of studies on selecting *a subset of* arms so that the total contribution of the selected arms is *close to a certain target.*[1]

Learning-based demand response. In addition to the demand response program considered in this paper and in Edison (2019), Lesage-Landry and Taylor (2017), PSEG (2019), ThinkEco (2019) and Wang et al. (2014), where customers are directly selected by the aggregator to perform demand response, there is a different type of DR programs based on dynamic pricing, where the goal is to design time-varying electricity prices to automatically incentivize desirable load reduction behaviors from the consumers (Faruqui, Sergici, & Palmer, 2010). Learning-based algorithms are also proposed for this type of DR programs to deal with, for example, the unknown utility functions of the consumers (Khezeli & Bitar, 2017; Li, Wang, & Zhang, 2017; Moradipari, Silva, & Alizadeh, 2018).

Preliminary work. Some preliminary work was presented in the conference paper (Li, Hu, & Li, 2018). This journal version strengthens the regret bounds, especially for the time-varying target case, conducts more intensive numerical analysis using realistic data from ISOs, provides more complete proofs, and adds more intuitions and discussions to 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, we define $A - B := \{x \mid x \in A, x \notin B\}$. When $k = 0$, let $\sum_{i=1}^{k} a_i = 0$ for any a_i, and define the set $\{\sigma(1), \ldots, \sigma(k)\} = \emptyset$ for any $\sigma(i)$. For $x \in \mathbb{R}^k$, we consider $\|x\|_\infty = \max_{i\in[k]} |x_i|$, and write $f(x) = O(g(x))$ as $\|x\|_\infty \to +\infty$ if there exists a constant M such that $|f(x)| \le M|g(x)|$ for any x with $\|x\|_\infty \ge M$; and $f(x) = o(g(x))$ if $f(x)/g(x) \to 0$ as $\|x\|_\infty \to +\infty$. We usually omit "as $\|x\|_\infty \to +\infty$" for simplicity. For the asymptotic behavior near zero, we define it by letting the inverse of $\|x\|_\infty$ go to infinity.

2. Problem formulation

Motivated by the discussion above, we formulate the demand response (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 DR program with an aggregator and n residential customers over T time steps, where each time step corresponds

[1] In our online supplementary file (Li, Hu, & Li, 2020), we provide an algorithm based on the risk-aversion MAB ideas and provide numerical results.

to one DR event.[2] Each customer is viewed as an arm in our CMAB problem. We consider a simple user (customer) behavior model, where each customer may either 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 $1 - p_i$. We denote the demand reduction by customer i at time step t as $X_{t,i}$, which is assumed to follow Bernoulli distribution, $X_{t,i} \sim \text{Bern}(p_i)$, and is independent across time.[3] Different customers behave independently and may respond to the same DR event with different probabilities. Though this behavior model may be oversimplified by neglecting the influences of temperatures, humidities, user fatigue, changes in lifestyles, etc., this simple model allows us to provide useful insights on improving the reliability of the DR programs and lay the foundation for future research on more realistic behavior models.

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

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

Due to the randomness of the demand reduction $X_{t,i}$, we aim to select a subset of customers S_t to minimize the squared deviation in expectation, that is,

$$S_t^* = \arg\min_{S_t \subseteq [n]} \mathbb{E}\left[L_t(S_t) \right]. \tag{1}$$

When there are multiple optimal solutions to (1), S_t^* is defined as any one of the optimal solutions.

In this paper, we will first study the scenario where the target D is time-invariant (Sections 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).

When the response probability profile $p = (p_1, \ldots, p_n)$ is known, the problem (1) is a combinatorial optimization. In Section 3, we will provide an offline combinatorial optimization algorithm to solve the problem (1).

In reality, the response probabilities are usually unknown. Thus, the aggregator should learn the probabilities from the feedback of the 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 Regret(T), which compares the total expected cost of online decisions and the optimal total expected costs in T time steps[4]:

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

where $R_t(S_t) := L_t(S_t) - L_t(S_t^*)$ 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 response of every selected customer, i.e. $\{X_{t,i}\}_{i \in S_t}$. Such feedback structure is called *semi-bandit* in literature (Chen et al., 2016), 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 provided below.

Example 1. Consider a crowd-sourcing related problem. Given a 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 intends 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,

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

Since the participation probabilities are unknown, the planner can learn the participation probabilities from the previous actions of the 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 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 and $D_t = D$, the problem (1) becomes a combinatorial optimization problem:

$$\min_{S \subseteq [n]} \left[\left(\sum_{i \in S} p_i - D \right)^2 + \sum_{i \in S} p_i (1 - p_i) \right]. \tag{3}$$

Though general combinatorial optimization is NP-hard and only has approximate algorithms, we introduce our Algorithm 1 which can 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 $\phi(p, D) \subseteq [n]$. The next theorem shows that Algorithm 1 outputs 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:
$p_{\sigma(1)} \geq \ldots \geq p_{\sigma(n)}$.
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: **Outputs**: $\phi(p, D) = \{\sigma(1), \ldots, \sigma(k)\}$

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

[3] 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 sizes 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 (Edison, 2019).

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

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 our supplementary material (Li et al., 2020) and only introduce the intuition here. An optimal set S should enjoy two properties: (i) the total expected contribution of S, i.e. $\sum_{i\in S} p_i$, 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 a 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 a 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), i.e.

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

where $J_i(t)$ denotes the set of time steps when arm i is selected by time t (including t) and $T_i(t) = |J_i(t)|$ 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.

3.3. Two simple online algorithms: Greedy algorithm and CUCB

Next, we introduce two simple methods: greedy algorithm and CUCB, and explain their poor performance in our problem to gain intuitions for our algorithm design.

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 that generate Bernoulli rewards with expectation $p_1 > p_2 > 0$. The goal is to select the arm with the higher reward in expectation, which is arm 1 in this case. Suppose after some time steps, arm 1's history sample average $\bar{p}_1(t)$ is zero, while arm 2's history average $\bar{p}_2(t)$ is positive. In this case, the greedy algorithm will always select the suboptimal arm 2 in the future since $\bar{p}_2(t) > \bar{p}_1(t) = 0$ for all future time t and arm 1's history average will remain 0 due to insufficient exploration. Hence, the regret will be linear with T.

A well-known algorithm in CMAB literature that balances the exploration and exploitation is CUCB (Chen et al., 2016). Instead of using sample average $\bar{p}(t-1)$ directly, CUCB considers an upper confidence bound:

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

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 Lemma 5. 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

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:** α, D.
3: **Initialization:** For $t = 1, \ldots, \left\lceil \frac{n}{\lceil 2D\rceil} \right\rceil$, 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 $T_i(t)$ and $\bar{p}_i(t)$ by the observation $\{X_{t,i}\}_{i\in S_t}$.[5]
4: **for** $t = \left\lceil \frac{n}{\lceil 2D\rceil} \right\rceil + 1, \ldots, T$ **do**
5: Compute the upper confidence bound for each i
$U_i(t) = \min\left(\bar{p}_i(t-1) + \sqrt{\frac{\alpha \log t}{2T_i(t-1)}}, 1\right)$.
6: Rank $U_i(t)$ by a non-increasing order:
$U_{\sigma(t,1)}(t) \geq \ldots \geq U_{\sigma(t,n)}(t)$.
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 let $k_t = n$ if $\sum_{i=1}^{n} \bar{p}_{\sigma(t,i)}(t-1) \leq D - 1/2$.
8: Select $S_t = \{\sigma(t,1), \ldots, \sigma(t,k_t)\}$
9: Update $T_i(t)$ and $\bar{p}_i(t)$ by observations $\{X_{t,i}\}_{i\in S_t}$
10: **end for**

3.4. Our proposed online algorithm: CUCB-Avg

Based on the discussion above, we propose a new method CUCB-Avg. The novelty of CUCB-Avg is that it utilizes both sample averages and upper confidence bounds by exploiting the structure of the offline 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 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 have a small $T_i(t-1)$, meaning that the arm has not been explored enough; or have 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*).

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

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. 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*

$$\text{Regret}(T) \le M\left(\left\lceil \frac{n}{\lceil 2D \rceil}\right\rceil + \frac{2n}{\alpha - 2}\right) + \frac{\alpha M n \log T}{2\epsilon_0^2}, \tag{5}$$

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, indicating that our algorithm learns the customers' response probabilities effectively. The dependence on n is polynomial, i.e. $O(n^3)$ by $M \sim O(n^2)$, showing that our algorithm can handle a large number of arms effectively. The cubic dependence is likely to be a proof artifact and improving the dependence on n is left as future work.

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 Li et al. (2020) and only explain the intuition behind ϵ_0 here. Roughly, ϵ_0 is a robustness measure of our offline optimal algorithm, in the sense that if the probability profile p is perturbed by ϵ_0, i.e., $|\tilde{p}_i - p_i| < \epsilon_0$ for all i, the output $\phi(\tilde{p}, D)$ 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 simplifying assumptions in the following proposition.

Proposition 1. *Consider the following assumptions.*
(A1) p_i are positive and distinct $p_{\sigma(1)} > \cdots > p_{\sigma(n)} > 0$.
(A2) There exists $k \ge 1$ 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:

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

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 (Li et al., 2020) 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, we show if $\forall\, i,\ |\tilde{p}_i - p_i| < \epsilon_0$, then $\phi(\tilde{p}, D) = \phi(p, D)$. This can be proved in two steps. Step 1: when $\epsilon_0 \le \frac{\Delta_k}{2}$, the k arms with higher $\tilde{p}_i$ are the same k arms with higher p_i because for any $1 \le i \le k$ and $k+1 \le j \le n$, we have $\tilde{p}_{\sigma(i)} > p_{\sigma(k)} - \epsilon_0 \ge p_{\sigma(k+1)} + \epsilon_0 > \tilde{p}_{\sigma(j)}$. Step 2: by $\epsilon_0 \le \min\left(\frac{\delta_1}{k}, \frac{\delta_2}{k}\right)$ and the definition of δ_1 and δ_2, when $|\tilde{p}_i - p_i| < \epsilon_0$ for all i, we have $\sum_{i=1}^{k} \tilde{p}_{\sigma(i)} > D - 1/2$ and $\sum_{i=1}^{k-1} \tilde{p}_{\sigma(i)} < D - 1/2$. Consequently, by Algorithm 1, $\phi(\tilde{p}, D) = \{\sigma(1), \ldots, \sigma(k)\} = \phi(p, D)$.

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. $\{\Delta_i | \Delta_i > 0,\ 1 \le i \le n-1\}$. 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 ensure the total expected contribution below $D - 1/2$. An explicit expression of ϵ_0 is provided in our report (Li et al., 2020).

Comparison with the regret bound of classic CMAB. In classic CMAB literature when the goal is to select K arms with the highest parameters given a fixed integer K, the regret bound usually depends on $\frac{\Delta_K}{2}$ (Kveton et al., 2015). 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 $\tilde{p}$ with estimation error at most $\frac{\Delta_K}{2}$: $\forall\, i,\ |\tilde{p}_i - p_i| < \frac{\Delta_K}{2}$, the top K arms with the profile $\tilde{p}$ 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 with $1/\Delta_K$, 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.

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:

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

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:

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

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 event 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 event $\bar{E}_t$ and $B_t(\epsilon_0)$ 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 event $\bar{E}_t$ and $\bar{B}_t(\epsilon_0)$ happen,* the regret is zero due to the enough exploration of the selected arms (Lemma 3).

Notice that the time steps are not divided sequentially here. For example, it is possible that $t = 10$ and $t = 30$ belong to Part 2 while $t = 20$ belongs to Part 3.

Proof details: Firstly, it is without loss of generality to require $D \ge 1/2$ because when $D < 1/2$, the optimal set is known to be the empty set by Corollary 1, so the regret is zero by selecting no customers.

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

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

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

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*)**.** *Consider i.i.d. random variables $X_1, \ldots, X_m$ with support $[0, 1]$ and mean μ, then we have*

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

Lemma 1. *When $\alpha > 2$, we have*

$$\mathbb{E}\left[\sum_{t=1}^{T} I_{E_t} R_t(S_t)\right] \leq \frac{2Mn}{\alpha - 2}.$$

Proof. The number of times E_t happens is bounded by

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

where the first inequality is by enumerating possible $i \in [n]$, the second inequality is by enumerating possible values of $T_i(t-1)$: $\{1, \ldots, t-1\}$, the third inequality is by Chernoff–Hoeffding's inequality, and the last inequality is by $\sum_{t=1}^{T} \frac{1}{t^{\alpha-1}} \leq \int_1^{+\infty} \frac{1}{t^{\alpha-1}} \leq \frac{1}{\alpha-2}$. 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 $\mathbb{E}\left[\sum_{t=1}^{T} R_t(S_t) I_{\{\bar{E}_t, B_t(\epsilon_0)\}}\right] \leq \frac{\alpha M n \log T}{2\epsilon_0^2}$*

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 $1 \leq t \leq T$, if $\bar{E}_t$ and $\bar{B}_t(\epsilon_0)$ happen, CUCB-Avg selects an optimal subset and $\mathbb{E}\left[R_t(S_t) I_{\{\bar{E}_t, \bar{B}_t(\epsilon_0)\}}\right] = 0$. Consequently, the regret in Part 4 is 0.*

Proof Sketch: We defer the proof to Li et al. (2020) 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

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

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

To obtain the intuition for Fact 2, we consider the expression of ϵ_0 in (6) under Assumption (A1) (A2) in Proposition 1. Let $\phi(p, D) = \{\sigma(1), \ldots, \sigma(k)\}$ 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 $\epsilon_0 \leq \frac{\Delta_k}{2}$, we can show that the selected subset S_t is either a superset or a subset of the optimal subset $\{\sigma(1), \ldots, \sigma(k)\}$.

(ii) By $\epsilon_0 \leq \delta_1/k$, 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 $\{\sigma(1), \ldots, \sigma(k)\}$ is still larger than $D - 1/2$.

(iii) By $\epsilon_0 \leq \delta_2/k$, 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 $\{\sigma(1), \ldots, \sigma(k)\}$ 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 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 $0 < D_t \leq \bar{D}, \ \forall\ 1 \leq t \leq T$.

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

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

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

Before the proof, we make a few comments below.

Dependence on T. The bound is sublinear in T, i.e. $O(\sqrt{T \log T})$, indicating that our algorithm learns the users' 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 parameters 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 our algorithm will find the optimal subset and generate no more regret eventually. 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 may not find the optimal subset with high probability even when t is large. This roughly explains why the bound of the time-varying case is worse than that of the static case.

In addition, we provide some intuitive explanation for the scaling $O(\sqrt{T\log T})$. It can be shown that the regret at time t is almost bounded by the estimation error at time t under some conditions (Lemma 5). Since the estimation error is roughly captured by our confidence interval in (4), which scales like $O(\sqrt{\log T/t})$, the total regret scales like $\sum_{t=1}^{T} O(\sqrt{\log T/t}) = O(\sqrt{T\log T})$.

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: $O(n^3)$ by $\bar{M} \sim O(n^2)$, demonstrating that our algorithm can learn a large number of arms effectively in the time-varying case. Improving the cubic dependence on n is left as future work.

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 parts of the proof is similar to the static case. We also consider $D_t \geq 1/2$ without loss of generality due to Corollary 1. Besides, we also divide the time steps into four parts and complete the proof by summing up the regret bound of each part. The first three parts can be bounded in the same ways as the static case. The major difference comes from the 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 we replace 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 $\bar{E}_t$ and $B_t(\epsilon_1)$ happen: notice that Lemma 2 still holds so $\mathbb{E}\sum_{t=1}^{T} R_t(S_t) I_{\{\bar{E}_t, B_t(\epsilon_1)\}} \leq \frac{\alpha M n \log T}{2\epsilon_1^2}$.

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

Lemma 4. *The regret in Part 4 can be bounded by*

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

Proof. Our proof relies on the following lemma which shows that the regret at time t can roughly be bounded by the estimation error ϵ at t when $\epsilon \leq \epsilon_1$.

Lemma 5. *For any time step t, consider any D_t and any $0 < \epsilon \leq \epsilon_1$ such that $\mathbb{P}(\bar{E}_t, \bar{B}_t(\epsilon)) > 0$. Let $\mathcal{F}_t$ denote the natural filtration up to time t. For any $\mathcal{F}_{t-1}$ such that $\bar{E}_t$ and $\bar{B}_t(\epsilon)$ are true, we have $\mathbb{E}[R_t(S_t) \mid \mathcal{F}_{t-1}] \leq 2n\epsilon$.*

Proof sketch. Due to space limits, we defer the proof to Li et al. (2020) and discuss the proof ideas here. Firstly, we can show that under $\bar{E}_t$ and $\bar{B}_t(\epsilon)$, 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 utilizes the quadratic structure of the loss function. □

Provided with Lemma 5, we can prove Lemma 4. 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 $q = 0, 1, 2, \ldots$:

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

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:

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t(S_t) I_{\bar{E}_t \cap \bar{B}_t(\epsilon_1)}\right] = \sum_{q=1}^{T}\sum_{t=1}^{T} \mathbb{E}\left[R_t(S_t) I_{(H_t^{q-1} - H_t^q)}\right]$$

$$\leq \sum_{q=1}^{T}\sum_{t=1}^{T} 2n\eta_q \mathbb{P}(H_t^{q-1} - H_t^q)$$

$$\leq \sum_{q=1}^{T} 2n^2\eta_q = 2n^2 \sum_{q=1}^{T} \sqrt{\frac{\frac{\alpha\log T}{2}}{q - 1 + \frac{\alpha\log T}{2\epsilon_1^2}}}$$

$$\leq 2n^2\sqrt{\frac{\alpha\log T}{2}} \int_0^T \sqrt{\frac{1}{q - 1 + \frac{\alpha\log T}{2\epsilon_1^2}}}\, \mathrm{d}q$$

$$\leq 4n^2\sqrt{\frac{\alpha\log T}{2}}\sqrt{T + \frac{\alpha\log T}{2\epsilon_1^2}},$$

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 taking conditional expectation on $H_t^{q-1} - H_t^q$ and by $(H_t^{q-1} - H_t^q) \subseteq \bar{E}_t \cap \bar{B}_t(\eta_q)$ and $\eta_q \leq \epsilon_1$ and Lemma 5; the second inequality is because $H_t^{q-1} - H_t^q \subseteq \bigcup_{i=1}^{n}\{i \in S_t,\ T_i(t-1) = \frac{\alpha\log T}{2\epsilon_1^2} + q\}$ and $\{i \in S_t,\ T_i(t-1) = \frac{\alpha\log T}{2\epsilon_1^2} + q\}$ occurs at most once in T stages for each $i \in [n]$; the third inequality uses the fact that $T > 2$ and thus $\frac{\alpha\log T}{2\epsilon_1^2} > 1$. □

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 CUCB (Chen et al., 2016), which is briefly explained in Section 3.3, and Thompson sampling (TS), an algorithm with good empirical performance in classic MAB problems. In TS, the unknown parameter profile 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 Russo et al. (2017).

In our experiment, we consider a residential demand response program with 3000 customers. Each customer can either participate in the DR event by reducing 200 W 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*

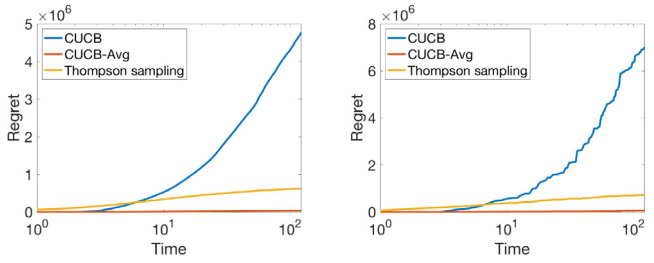


Fig. 1. The regret of CUCB, CUCB-Avg, and TS.

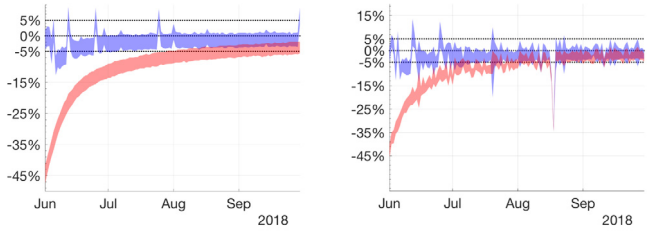


Fig. 2. 90% confidence intervals of load reduction's relative errors of CUCB-Avg (blue) and Thompson sampling (red).

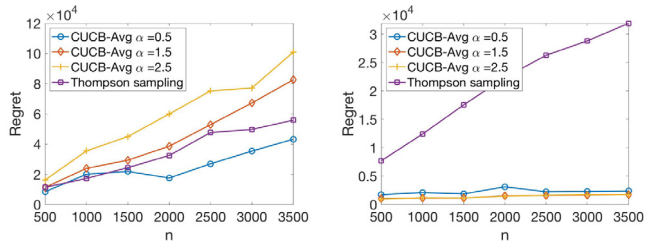


Fig. 3. The regret of TS and CUCB-Avg for different n.

demand profile is from New England ISO.[6] We consider two schemes to determine the peak-load-shaving target D_t:

(i) Average peak: Compute the averaged load profile in a day by averaging the daily load profiles in the four months. The constant target D is the 1% of the difference between the peak load and the load at one hour before the peak hour of the averaged load profile.

(ii) Daily peak: On each day t, the target D_t is 1% of the difference between the peak load and the load at one hour before the peak hour of the daily demand.

In our algorithms, we set $\alpha = 2.5$. In Thompson sampling, p's prior distribution is Unif$[0, 1]^n$. We consider one DR event per day and plot the daily performance.

Fig. 1 plots the regret of CUCB, CUCB-Avg and TS under the two schemes of peak shaving. The x-axis is in log scale and the resolution is by day. Both figures show that CUCB-Avg performs better than CUCB and TS. In addition, the regret of CUCB-Avg in Fig. 1(a) is linear with respect to $\log(T)$, consistent with our theoretical result in Theorem 2. Moreover, the regret of CUCB-Avg in Fig. 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.

Fig. 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 $\pm 5\%$, 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 too high to reach even after selecting all the users.

Finally, we compare TS and CUCB-Avg for different n by considering the scheme (i). We consider two cases: (1) when T is very large so the regret is dominated by the $\log(T)$ term, (2) when T is a reasonable number in practice. We let $T = 10^5$ for case 1 and $T = 122$ (the total number of days from June to September) for case 2. We consider a smaller target $D = 40$ for illustration and consider $n = 500 : 500 : 3500$. Fig. 3(a) shows that the dependence on n of CUCB-Avg's regret is similar to that of TS when T is large, and the dependence is not cubic, the theoretical explanation of which is left for future work. Moreover, Fig. 3(a) shows that CUCB-Avg can achieve better regrets than TS under a properly chosen small α. Though not explained by theory yet, the phenomenon that a small α yields good performance has been observed in literature (Wang & Chen, 2018). Further, Fig. 3(b) shows that CUCB-Avg achieves significantly smaller regrets than TS for a practical T, indicating the effectiveness of our algorithm in reality.

6.2. More discussion on the effects of α and n

Fig. 3 has shown that the choices of α and n affect the algorithm performance. In this subsection, we will discuss the effects of α and n in greater details. In particular, we will study the DR performance by the relative deviation of the load reduction, which is defined as $\sqrt{\mathbb{E}[L(S_t)]}/D_t$, for each day during the four months.

Fig. 4 shows the relative deviation of CUCB-Avg for different α when $n = 3000$ and when the target is determined by scheme (i) in Section 6.1. It is observed that when T is small, a large α provides smaller relative deviation, thus better performance. This is because the information of customers is limited when T is small, and larger α encourages exploration of the information, thus yielding better performance. When T is large, a smaller α leads to better performance most of the time. 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. The observations above are also consistent with Fig. 3. Further, Fig. 4 shows that for a wide range of α's values, CUCB-Avg reduces the deviation to below 5% after a few days, indicating that CUCB-Avg is reasonably robust to the choice of α.

Fig. 5 shows the relative deviation of CUCB-Avg for different n when $\alpha = 2.5$ and when the target is determined by scheme (i) in Section 6.1. It is observed 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 are 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. It is

[6] https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/demand-by-zone.

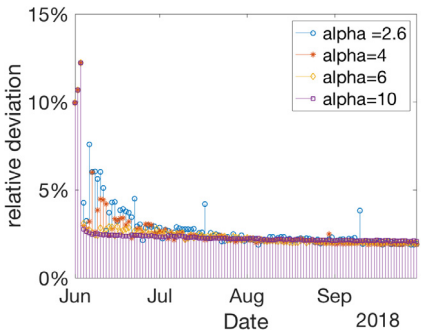


Fig. 4. Effects of α.

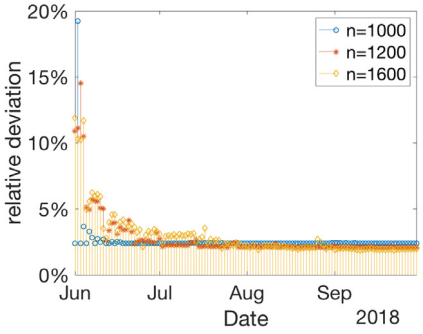


Fig. 5. Effects of n.

worth mentioning that though Fig. 3(a) shows that the regret increases with n when T is large, there is no conflict because the regret captures the gap between the deviation generated by the algorithm and the optimal one, which may increase even when the algorithm generates less deviation since the optimal deviation also decreases.

6.3. On the user fatigue effect

It is widely observed that customers tend to be less responsive to demand response signals after participating in DR events consecutively. This effect is usually called *user fatigue*. Though our algorithm and theoretical analysis do not consider this effect for simplicity, our CUCB-Avg can handle the fatigue effect after small modifications, which is briefly discussed below.

For illustrational purposes, we consider a simple model of the user fatigue effect. Each customer i is associated with an original response probability p_i. The response probability at stage t, denoted as $p_i(t)$, decays exponentially with a fatigue ratio f_i if customer i has been selected consecutively, that is, $p_i(t) = (f_i)^{\chi_i(t)} p_i$ if customer i has been selected from day $t - \chi_i(t)$ to day $t - 1$. If the customer is not selected , we consider that the customer takes a rest at this stage and will respond to the next DR event with the original probability. Though the fatigue model may be too pessimistic about the effects of the consecutive selections by considering exponential decaying fatigue factors, and too optimistic about the effectiveness of rests by assuming full recovery after one day rest, this model captures the commonly observed phenomena that the consecutive selection is a key reason for user fatigue and customers can recover from fatigue if not selected for some time (Hopkins & Whited, 2017). The model can be revised to be more complicated and realistic, which is left as future work.

Next, we explain how to modify CUCB-Avg to address the user fatigue effects. We consider that the aggregator has some initial estimation of the fatigue ratio of customer i, denoted as $\tilde{f}_i$, and will use the estimated fatigue ratios to rescale the upper confidence bounds and sample averages in Algorithm 2 to account for the fatigue effect. In particular, the rescaled upper confidence bound is $(\tilde{f}_i)^{\chi_i(t)} U_i(t)$, and the rescaled history sample average is $\bar{p}_i(t) = \frac{1}{T_i(t)} \sum_{\tau \in I_i(t)} \frac{X_{\tau,i}}{(\tilde{f}_i)^{\chi_i(t)}}$, where $\chi_i(t)$ denotes the number of consecutive days up until $t-1$ when customer i is selected.

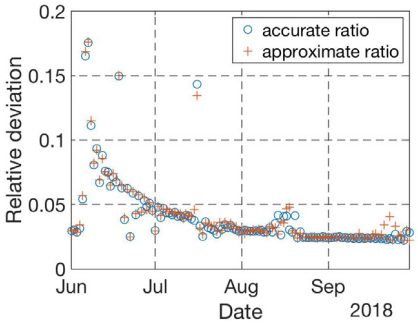


Fig. 6. The performance of our CUCB-Avg (after simple modifications) when considering user fatigue.

In our numerical experiments, different users may have different user fatigue ratios, which are generated i.i.d. from Unif [0.75, 0.95]. Other parameters are the same as in Section 6.1. Fig. 6 plots the relative deviation of our modified CUCB-Avg in two scenarios: (i) the aggregator has access to the accurate fatigue ratio, i.e. $\tilde{f}_i = f_i$; (ii) the aggregator only has a rough estimation for the entire population: $\tilde{f}_i = 0.85$ for all i. It can be observed that our algorithm is able to reduce the relative deviation to below 5% after a few days even when the fatigue ratios are inaccurate. This demonstrates that our algorithm, with some simple modifications, can work reasonably well even when considering customer fatigue effects.

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. There are several interesting directions to explore in the future. First, it is interesting to improve the dependence on n. Second, it is worth studying the regret lower bounds. Besides, it is worth considering more realistic behavior models which may include e.g. the effects of temperatures and humidities, the user fatigue, correlation among users, time-varying response patterns, general load reduction distributions, dynamic population, etc.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.automatica.2020.109015 and https://arxiv.org/pdf/2003.09505.pdf.

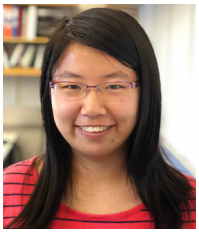
References

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*(2–3), 235–256.

Belleflamme, P., Lambert, T., & Schwienbacher, A. (2014). Crowdfunding: Tapping the right crowd. *Journal of Business Venturing*, *29*(5), 585–609.

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

Chen, W., Wang, Y., Yuan, Y., & Wang, Q. (2016). Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research (JMLR)*, *17*(1), 1746–1778.
Edison, C. (2019). Consolidated Edison smart AC program. https://conedsmartac.com.
Faruqui, A., Sergici, S., & Palmer, J. (2010). The impact of dynamic pricing on low income customers. https://www.edisonfoundation.net/IEE/Documents/IEE_LowIncomeDynamicPricing_0910.pdf.
FERC (2017). *Reports on demand response and advanced metering*: *Technical report*, Federal Energy Regulatory Commission.
Gopalan, A., Mannor, S., & Mansour, Y. (2014). Thompson sampling for complex online problems. In *International conference on machine learning* (pp. 100–108).
Hopkins, A. S., & Whited, M. (2017). Best practices in utility demand response programs. https://www.synapse-energy.com/sites/default/files/Utility-DR-17-010.pdf.
Jain, S., Narayanaswamy, B., & Narahari, Y. (2014). A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *AAAI* (pp. 721–727).
Khezeli, K., & Bitar, E. (2017). Risk-sensitive learning and pricing for demand response. *IEEE Transactions on Smart Grid*, *9*(6), 6000–6007.
Kuderer, M., Gulati, S., & Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 2641–2646). IEEE.
Kveton, B., Wen, Z., Ashkan, A., & Szepesvari, C. (2015). Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial intelligence and statistics* (pp. 535–543).
Lesage-Landry, A., & Taylor, J. A. (2017). The multi-armed bandit with stochastic plays. *IEEE Transactions on Automatic Control*.
Li, Y., Hu, Q., & Li, N. (2018). Learning and selecting the right customers for reliability: A multi-armed bandit approach. In *2018 IEEE conference on decision and control (CDC)* (pp. 4869–4874). IEEE.
Li, Y., Hu, Q., & Li, N. (2020). A reliability-aware multi-armed bandit approach to learn and select users in demand response. Supplementary material, https://arxiv.org/abs/2003.09505.
Li, Y., & Li, N. (2017). Mechanism design for reliability in demand response with uncertainty. In *2017 American control conference (ACC)* (pp. 3400–3405). IEEE.
Li, P., Wang, H., & Zhang, B. (2017). A distributed online pricing strategy for demand response programs. *IEEE Transactions on Smart Grid*, *10*(1), 350–360.
Moradipari, A., Silva, C., & Alizadeh, M. (2018). Learning to dynamically price electricity demand based on multi-armed bandits. In *2018 IEEE global conference on signal and information processing (GlobalSIP)* (pp. 917–921). IEEE.
O'Neill, D., Levorato, M., Goldsmith, A., & Mitra, U. (2010). Residential demand response using reinforcement learning. In *2010 first IEEE international conference on smart grid communications (SmartGridComm)* (pp. 409–414). IEEE.
PSEG (2019). Cool customer program. https://nj.pseg.com/saveenergyandmoney/energysavingpage/coolcustomerprogram.
Russo, D., Van Roy, B., Kazerouni, A., & Osband, I. (2017). A tutorial on thompson sampling. arXiv preprint arXiv:1707.02038.
Sani, A., Lazaric, A., & Munos, R. (2012). Risk-aversion in multi-armed bandits. In *Advances in neural information processing systems* (pp. 3275–3283).
ThinkEco (2019). Smart AC program. http://www.thinkecoinc.com/#smart-control.
Vakili, S., & Zhao, Q. (2016). Risk-averse multi-armed bandit problems under mean-variance measure. *IEEE Journal of Selected Topics in Signal Processing*, *10*(6), 1093–1111.
Wang, S., & Chen, W. (2018). Thompson sampling for combinatorial semi-bandits. In *International conference on machine learning* (pp. 5101–5109).
Wang, Q., Liu, M., & Mathieu, J. L. (2014). Adaptive demand response: Online learning of restless and controlled bandits. In *Smart grid communications (SmartGridComm), 2014 IEEE international conference on* (pp. 752–757). IEEE.



Yingying Li received her B.S. degree in Mathematics and Applied Mathematics from University of Science and Technology of China in Hefei, China in 2015. Since 2015, she has been a graduate student in the John A. Paulson School of Engineering and Applied Sciences, Harvard University. Her research interest lies in online optimization, online optimal control, reinforcement learning and multi-armed bandits.

Qinran Hu received the B.S. degree from Chien-Shiung Wu Honors College, Southeast University (China) in 2010, and the M.S. and Ph.D. degrees from the University of Tennessee, Knoxville, TN, USA in 2013 and 2015, respectively. He was a postdoc fellow in Harvard University, Cambridge, MA, USA from 2015 to 2018. He joined the School of Electrical Engineering, Southeast University, Nanjing, China in Oct. 2018. His research interests include power system operation optimization and demand aggregation.

Na Li received her B.S. degree in Mathematics and Applied Mathematics from Zhejiang University in China and her PhD degree in Control and Dynamical systems from the California Institute of Technology in 2013. She is currently a Thomas D. Cabot Associate Professor in the School of Engineering and Applied Sciences at Harvard University where she joined in 2014 as an assistant professor. She was a postdoctoral associate of the Laboratory for Information and Decision Systems at Massachusetts Institute of Technology. She won NSF CAREER Award and AFOSR YIP Award. Her research lies in the design, analysis, optimization and control of distributed network systems, with particular applications to power networks and systems biology/physiology.