Online learning 0970249
Cost-Aware Cascading Bandit

Dovid Parnas 337977045, Aharon Renick 015782899

January 2025

Selected paper:
C. Gan, R. Zhou, J. Yang and C. Shen, "Cost-Aware Cascading Bandits", in IEEE Transactions on Signal Processing, vol. 68, pp. 3692-3706, 2020, doi: 10.1109/TSP.2020.3001388. [1]

Abstract

We review the paper Cost-Aware Cascading Bandits, describing a setting of cascading bandits with costs. Cascading bandits offer ranked lists of arms that are deployed until an arm "succeeds" or the list ends. The reviewed paper extends cascading bandits to cost-aware settings, where each arm has a cost for deploying it. This setting is unique since reward is dependent on entire lists of arms deployed, while cost is dependent on each single arm. The reviewed paper presents a UCB-based algorithm for this setting, with possible settings variations and corresponding algorithm adaptations. The reviewed paper proves lower and upper bounds for the presented algorithms, showing it is order optimal. In our extension, we offer an adaptation of the Cost-Aware Cascading Bandit setting to the best-arm identification paradigm, defining a new algorithm for this case, and use a simulation to demonstrate when this may be beneficial.

1 Introduction

1.1 Model

The paper proposes a new multi-armed bandit (MAB) model called the Cost-Aware Cascading Bandits (CCB) to address practical challenges in applications such as recommendation systems. The model presented includes a stochastic bandit with K-armed Bernoulli rewards. The model extends standard MAB in three main ways:

Cascading Feedback: A ranked list of arms is presented to the learning agent. The agent pulls arms as a sequence, collecting feedback until an arm returns a reward, resulting in partial and sequential observability.

Action Costs: Selecting each arm incurs a stochastic cost from an unknown, non-negative distribution. The learner objective is to maximize the total net

reward (rewards minus costs) over time, which introduces trade-offs between potential reward and cost-efficiency. Two versions of Cost-Aware Cascading Bandits are offered, one for when costs are unknown at the beginning of each round and the other when costs become known at the beginning of each round.
Fail-Safe Fine: An optional extension to the model is presented in the paper, addressing a ”fail-safe” setting. If the agent does not receive a reward for the entire list of arms pulled, it is fined a set amount. This compels the agent to take on risk to avoid the ”fail-safe” fine. In turn, this changes which arms are worth-while for the agent to include in its list each round.

1.2 Key Findings

The authors propose the Cost-Aware Cascading Upper Confidence Bound (CC-UCB) algorithm, which is an adaptation of UCB to handle cascading feedback and cost in the unknown cost setting. The algorithm appears in Appendix 4.1. The crux of the algorithm is estimating the expected rewards and costs of the arms, using UCB to encode uncertainty, and ordering the arms by the ratio of expected reward to cost. They prove this is an optimal ordering.
They also propose slight modifications of the algorithm (CC-UCB2) for the known-cost setting and a modification for the ”fail-safe” setting. They prove regret bounds for their proposed variations of CC-UCB, showing it achieves log order regret (under some assumptions about reward and cost distributions). They also show that this is order-optimal regret performance. Numerical simulations are also used to demonstrate the performance of CC-UCB and CC-UCB2.

1.3 Methods

Reward and cost estimation: CC-UCB constructs confidence bounds for rewards and cost, similar to the UCB principal.
Cost awareness: CC-UCB incorporates cost into the decision-making process by extending reward estimates to include net gains, both for ordering the arms and for defining the stopping criteria.

1.4 Literature Review

UCB: The study builds on foundational work in stochastic MAB of Upper Confidence Bound (UCB) algorithms, widely used [2]. This is one of the key tools we learned in class.
Arm cost: Research integrating cost and budget constraints into the MAB framework can be categorized into two types: The first type presents a budget constraint on the total number of pulls, when the goal is to determine the best arm [3], [4]. The second type involves costs per arm pull constrained by a budget, with the objective of maximizing the total reward under these constraints, either for fixed arm-pulling costs [5], [6], or random costs [7], [8]. For single-arm pulls per step, costs can be absorbed into rewards, reducing the model to a conventional MAB setting. The setting introduced in the reviewed

paper introduces an approach where the cost is for each arm and reward for the total list of arms played each step, so the cost cannot be absorbed into the rewards.

Cascading Bandits: Multi armed bandits in a cascading bandit (CB) setting, where a sequence of arms are pulled each step have been previously researched, for example in [9], [10]. The CCB model presented in the reviewed paper shares similarities with cascading bandits (CB), where an agent predetermines an ordered list of arms, gaining a reward if at least one arm generates a reward. However, CB normally assumes a fixed number of arms per step and does not include costs, leading to identical expected rewards regardless of order. In CCB, the ranking of arms affects the expected net reward, thus the paper extends the possible offline policies and online algorithms. A cascading bandit setting with pure-exploration goal is presented in [11], but with the classical fixed number of arms per step and no cost, which we discuss in our extension.

2 Implementation - Proof Sketches

2.1 Definitions and Notations

Given a K-armed stochastic bandit where the state of each arm is independent between steps, we denote the random variable representing the reward from arm i at time step t as $X_{i,t} \sim Ber(\theta_i)$. Let $Y_{i,t}$ denote the cost of arm i at time t, a bounded, non-negative random variable with $\mathbb{E}[Y_{i,t}] = c_i$.

Because this is a cascading bandit, the learning agent chooses an ordered list each turn and pulls the arms sequentially until one of them provides a reward or it has iterated over all arms in the list. We will denote this list as $I_t := \{I_t(1), I_t(2), ..., I_t(|I_t|)\}$. We will note that if one of the arms pulled provides a reward then only a truncated list, $\tilde{I}_t$ will be experienced by the learning agent. The rewards and costs accrued each round depend on the actualization of the pulled arms. Pulling the sequence of arms will contribute a reward of one if at least one of the arms receives a reward of one. Each arm pulled will also add its cost. Thus, the net reward at step t, r_t, is the following

$$r_t := 1 - \prod_{i=1}^{|\tilde{I}_t|}(1 - X_{\tilde{I}_t(i),t}) - \sum_{i=1}^{|\tilde{I}_t|} Y_{\tilde{I}_t(i),t}$$

The regret at each step t is $\mathbf{reg_t} := r_t^* - r_t$.

The cumulative regret is then $R(T) := \mathbb{E}[\sum_{t=1}^{T} \mathbf{reg_t}]$.

Let us define an α**-consistent** policy. Given an online policy that sequentially pulls arms in I_t until rewarded or all arms are pulled, the policy is α**-consistent** if $\mathbb{E}[\sum_{t=1}^{T} \mathbb{1}(I_t \neq I^*)] = o(T^\alpha)$.

For several proofs later we will use, $p_i := \frac{\prod_{j=1}^{k}(1-\theta_j)}{(1-\theta_i)}$. This variable will be 1 if no arm was rewarded. Otherwise, it is zero.

Let the difference between the expected cost and reward of an arm be denoted $\Delta_i := c_i - \theta_i$.

In the algorithm and proofs, we use UCB padding, $u_{i,t}$, proportionate to α, the current round, t, and how many times an arm i has been pulled, $N_{i,t}$. The UCB padding is used to upper bound arms' reward and lower bound their cost. These are designated $U_{i,t}$ and $L_{i,t}$, respectively.
The reviewed paper offers several variations of the cost-aware cascading bandit for different cost settings.

1. Unknown Costs - Nothing is known about the costs beyond the information collected by the agent.
2. Known Immediate Costs - The agent receives at the beginning of each round the costs of each arm. These change between rounds.
3. Fail-Safe Cost - For either unknown costs or known immediate costs, a penalty is added if none of the arms proposed by the agent result in a reward.

For each of these settings, we summarize the optimal offline policy and the online algorithm with proofs for upper and lower regret bounds. Because the proofs are similar between the settings, we provide proof sketches for the unknown setting. For the other two settings, we point out differences to the unknown setting that affect the proof and also where the proofs differ from the unknown setting.

2.2 Unknown Costs

This section covers the case when costs are entirely unknown to the algorithm for all steps.
Optimal Offline Policy Assuming that $c_i > 0, \forall i$ and that the arm states and costs are i.i.d. across different steps, we find the optimal policy for a single step and then apply it at each step to obtain optimal regret. Additionally, in the offline setting the statistics are available to the decision maker.
The optimization problem may be formalized:

$$\arg\min_{I} \sum_{i=1}^{|I|} (\theta_{I(i)} - c_{I(i)}) \prod_{j=1}^{i-1} (1 - \theta_{I(j)})$$

If $\theta_i \neq c_i$, for all $i \in [K]$ and $\exists \epsilon > 0$ s.t. $c_i > \epsilon$ for all $i \in [K]$, then UCR-T1, Unit Cost Ranking with Threshold 1, is the optimal offline policy according to **Theorem 1**.
UCR-T1 orders the arm indices such that L^* arms are pulled and the order is set by:

$$\frac{\theta_{1^*}}{c_{1^*}} \geq ... \geq \frac{\theta_{L^*}}{c_{L^*}} > 1 > \frac{\theta_{(L+1)^*}}{c_{(L+1)^*}} \geq ... \geq \frac{\theta_{K^*}}{c_{K^*}}$$

Theorem 1 compares the reward given $I^* = (1^*, 2^*, ..., L^*)$ with an alternative ordering which changes the order between two entries, i and $i+1$, s.t. $\frac{\theta_{i+1}}{c_{i+1}} > \frac{\theta_i}{c_i}$.

They isolate the terms that disagree and show that the reward for I^* is larger than the alternative.
Lastly, if $\frac{\theta_i}{c_I} < 1$ then in expectation the cost is higher than the reward and pulling arm i will not result in optimal regret.
Online Policy Algorithm The authors present ***Algorithm 1:*** *Cost-Aware Cascading UCB (CC-UCB)* for the online case when costs are unknown *a priori*. They continue the line of thought from the offline policy, using the proportion of the expected reward to cost to order arms. They use UCB padding to create proportions of an approximated upper bound on the reward to an approximated lower bound of the cost. The CC-UCB algorithm appears in Appendix 4.1. We sketch the proofs for its upper and lower bounds. We begin with the briefer lower bound.
Online Policy Algorithm - Lower Bound
They show that given **CC-UCB**, the cumulative regret is lower bounded according to **Theorem 3** by:

$$\lim_{T\to\infty} \inf \frac{R(T)}{\log T} \geq \sum_{i\in[K]\setminus I^*} \frac{(c_i-\theta_i)}{d(\theta_i;c_i)}$$

We will outline the proof of the lower bound:
Lower Bounding Stepwise Regret - First, we lower bound the regret in a single step. **Lemma 5** claims that the expected regret at any step is lower bounded by the expected difference between costs and rewards incurred by pulling non-ideal arms. *Lemma 5* first lower bounds the regret by the difference between the actual reward and the reward of the ideal arms assigned to be pulled at step t. We then consider two cases. Denoting $\tilde{I}_t\setminus I^*$ as the non-ideal arms set to be pulled, we label $\tilde{i}_t$ the last pulled non-ideal arm. If arm $\tilde{i}_t$ realized a reward, $X_{\tilde{i},t} = 1$, then the regret is at least the non-ideal costs minus the reward. Otherwise, if $X_{\tilde{i},t} = 0$, then the regret is at least the costs of pulling the non-ideal arms. If we reformulate this in terms of cumulative regret we get:

$$R(T) \geq \sum_{i\in[K]\setminus I^*} (c_i-\theta_i)\mathbb{E}[N_{i,T}]$$

That is to say that the regret is at least the difference between non-ideal arms' costs and the reward scaled by the expected number of steps that those arms were pulled. Now we must lower bound $\mathbb{E}[N_{i,T}]$ for each arm i.
Lower Bounding the number of steps when non-ideal arms were pulled - Assuming that the policy is $\alpha-$*consistent*, the authors construct an alternative instance distribution such that under that alternative any arm may be included in I^*. The number of times that such an arm will be pulled is lower bounded by $o(T^\alpha, \forall\alpha \in (0,1)$ due to $\alpha-$ *consistency*. This allows $N_{i,T}$ to meet the criteria necessary for an inequality found in [12]. The inequality is:

$$\lim_{T\to\infty} \inf \frac{\mathbb{E}[N_{i,T}]}{\log T} \geq \frac{1}{d(\theta_i;c_i)}$$

Together, *Lemma 5* and the lower bound on non-ideal arms pulled result in **Theorem 3**. The implication of this is that the regret under Algorithm 1 is $o(\log T)$.

Online Policy Algorithm - Upper Bound
The reviewed paper shows that given **CC-UCB** and $\alpha \geq 1.5$, the cumulative regret is upper bounded according to **Theorem 2** by:

$$R(T) \leq \sum_{i \in [K] \setminus I^*} c_i \frac{16\alpha \log(T)}{\Delta_i^2} + O(1)$$

The implication of this is that the regret under **CC-UCB** is $O(\log T)$. Together with the lower bound, we conclude that **CC-UCB** has order-optimal regret performance.
Theorem 2 decomposes the regret into three parts, corresponding to three contributing factors to the regret:

1. Misestimation of the empirical average
2. Incorrectly ordering the ideal set of arms
3. Pulling arms not in the ideal set of arms

Likewise, we will split the outline of the proof of **Theorem 2** into three primary parts based on this decomposition.

2.2.1 Misestimation of the empirical average

The authors begin by defining misestimation if either the reward ($\hat{\theta}_i$) or the cost ($\hat{c}_{i,t}$) are outside of the UCB padded intervals.

$$\mathcal{E}_t := \{\exists i \in [K], |\hat{\theta}_{i,t} - \theta_i| > u_{i,t} \text{ or } |\hat{c}_{i,t} - c_i| > u_{i,t}\}$$

The case when the sample averages are contained within the UCB bounds for all arms is the complement of $\mathcal{E}_t$, $\bar{\mathcal{E}}_t$.
Lemma 2 bounds the expected number of steps where there is misestimation, $\mathcal{E}_t$, under Algorithm 1 with $\alpha \geq 1.5$ with a constant.

$$\sum_{t=1}^{T} \mathbb{E}[\mathbb{1}(\mathcal{E}_t)] <= \psi := K(1 + \frac{4\pi^2}{3})$$

Therefore, this part of the regret contributes $O(1)$ to the overall regret.
We will outline the proof:
1. Initialization - The first time that each arm is pulled and observed is bounded by K steps. Each arm can contribute at most 1 to the expectation of steps with misestimation, resulting in an upper bound of K.
2. Limiting the sample error - Using union bound, we look at each arm

separately and bound the probability of an arm's reward or cost being outside of the UCB bound. This allows us to bound the probability of the sample error being larger than the UCB padding using Hoeffding's inequality. The resulting sum can be bounded using the Rieman-Zeta function, providing the constant $\frac{4\pi^2}{3}K$.
Thus, the expected number of steps with misestimation is bounded by a constant.

2.2.2 Incorrectly ordering the ideal set of arms

The authors define random variable $\mathcal{B}_t$ to describe a round with ideal arms incorrectly ranked. If I^* is the ideal ranking of the L ideal arms then:

$$\mathcal{B}_t := \{\exists i^*, j^* \in I^*, i < j, s.t. \frac{U_{i^*,t}}{L_{i^*,t}} < \frac{U_{j^*,t}}{L_{j^*,t}}\}$$

Lemma 3 bounds the expected number of steps where there is no misestimation but the ideal arms are incorrectly ordered by a constant. Under CC-UCB with $\alpha \geq 1.5$.

$$\mathbb{E}[\sum_{t=1}^{T} \mathbb{1}(\bar{\mathcal{E}}_t)\mathbb{1}(\mathcal{B}_t)] \leq \zeta$$

Therefore, this part of the regret contributes $O(1)$ to the overall regret.

We will outline the proof:
Reformulating steps with no misestimation - Two new definitions allow us to sum over only the steps when there was no misestimation, instead of all steps. First, τ_n is the minimal step such that $\sum_{t=1}^{\tau_n} \mathbb{1}(\bar{\mathcal{E}}_t) = n$. Second, $\Gamma_T := \sum_{t=1}^{T} \mathbb{1}(\bar{\mathcal{E}}_t)$. Therefore, the steps when there is no misestimation are the set $\{\tau_n, \forall n \in [\Gamma_T]\}$.
Reformulating incorrect ordering - Union bound is used to separate the indicators that each arm has been misordered. Then the indicator of the inequality expressing a misordering, $\frac{U_{j^*,t}}{L_{j^*,t}} > \frac{U_{(j-1)^*,t}}{L_{(j-1)^*,t}}$, is reorganized as an indicator bounding the number of times that each arm is pulled. We note that the summation of arms is over each sequential set of (j, j-1) arms in $j \in [L]\backslash\{1\}$ gives a total of $L-1$ indicators.
Combining the Two Reformulations - We replace the summation with an indicator for no misestimation with the sum over the set $\{\tau_n, \forall n \in [\Gamma_T]\}$. Denoting $\Delta_{i,j} := \frac{(\frac{\theta_i}{c_i} - \frac{\theta_j}{c_j})c_j}{2(1+\frac{\theta_j}{c_j})}$, we look at the expected probability of a misordered arm being number of times a bounded amount of times, $\mathbb{E}[\sum_{t=1}^{\Gamma_T} \mathbb{1}(N_{j^*,\tau_n} < \frac{\alpha \log \tau_n}{\Delta^2_{(j-1)^*,j^*}})]$.
Divide and Conquer - Now that we iterate only over steps with no misestimation, we bound the above term in two scenarios - when most steps do not have misestimation up to τ_n and when most steps do, $\mathbb{1}(\tau_n \leq 2n)$ or $\mathbb{1}(\tau_n > 2n)$. We

do so by bounding the term, $\frac{\alpha \log \tau_n}{\Delta^2_{(j-1)^*,j^*}}$. The two cases are expanded upon in the Appendix.

Combining the two cases - After dividing the steps without misestimation into two sets, $\mathbb{1}(\tau_n \leq 2n)$ and $\mathbb{1}(\tau_n > 2n)$, we combine the bounds on both of them and receive

$$\mathbb{E}[\sum_{t=1}^{T}]\mathbb{1}(\bar{\mathcal{E}}_t\mathbb{1})(\mathcal{B}_t))] \leq \sum_{j=2}^{L}(\zeta_j + \frac{1}{2p_{j^*}^2}) + \xi_0 := \zeta$$

2.2.3 Pulling non-ideal arms

Next, we bound the regret from pulling arms that are not in the ideal set of arms, $\tilde{I}_T \backslash I^*$. Because we have already bounded steps with misestimation and misordering of ideal arms, here we only bound the regret of pulling non-ideal arms in steps without misestimation or misordering. We prove that:

$$\mathbb{E}[\sum_{t=1}^{T}\mathbb{1}(\bar{\mathcal{E}}_t)\mathbb{1}(\bar{\mathcal{B}}_t)\mathbb{E}[reg_t|\tilde{I}_t]] \leq \sum_{i\in[K]\backslash I^*} c_i \frac{16\alpha \log T}{\Delta_i^2}$$

We outline the proof:

Bounding the regret caused by pulling non-ideal arms - Using **Lemma 4** (See Appendix), we bound the regret incurred in a given round due to pulling non-ideal arms to the cost of those arms.

Bounding the steps without misestimation and correct ordering of ideal arms - Using union bound, we examine each non-ideal arm, $[K]\backslash I^*$ and notice that when the ideal arms are correctly ordered is a sub-case of when a given non-ideal arm is included in I_t. Therefore, is is bounded by that case. We then reformulate the indicator for the case of no misestimation and a non-ideal arm i is included in I_t, that is to say $\frac{U_{i,t}}{L_{i,t}} > 1$ and manipulate the terms to limit the number of times that each arm i is pulled.

This concludes the proof.

2.3 Known Immediate Costs

This section covers the case when costs come from an unknown, bound, non-negative distribution and are known only for the immediate coming step. The algorithm sees $\{Y_{i,t}\}_{i\in[K]}$ preceding round t and the costs may change between rounds. Thus, the knowledge of the costs causes each round to have a different set of arms that should be pulled. The authors make three assumptions for this setting: the cost is never equal to the expected reward, the cost is strictly positive and the expected reward to cost ratio is unique per arm.

Optimal Offline Policy

Following the same logic as *Theorem 1*, **Theorem 4** claims that the optimal offline policy is to order the arms per step by their expected reward to immediate cost ratio, $Y_{i,t}$.

$$\frac{\theta_{1^*}}{Y_{1^*,t}} \geq ... \geq \frac{\theta_{L^*}}{Y_{L^*,t}} > 1 > \frac{\theta_{(L+1)^*}}{Y_{(L+1)^*,t}} \geq ... \geq \frac{\theta_{K^*}}{Y_{K^*,t}}$$

Online Policy Algorithm
The authors present CC-UCB2, which is identical to CC-UCB except that the expected cost, $\hat{c}_{i,t}$, is switched with known immediate cost, $Y_{i,t}$, in line 6 such that $L_{i,t} = \max(Y_{i,t} - u_{i,t}, \epsilon)$. See CC-UCB in the Appendix.

Online Policy Algorithm - Lower Bound
Using the same logic as *Lemma 5* in the unknown cost setting, **Lemma 9** claims that the expected regret at any step is lower bounded by the difference between the immediate costs and the rewards incurred by pulling non-ideal arms.
Theorem 6 uses *Lemma 9* to lower bound the cumulative regret by:

$$\lim_{T\to\infty} \inf \frac{R(T)}{\log T} \geq \sum_{i \in S_2} \frac{(c_i - \theta_i)}{d(\theta_i; c_i)}$$

The proof for *Theorem 6* differs from *Theorem 3* only slightly. The difference is caused by the non-existence of a single set of non-ideal arms. In the context of known immediate costs, each turn has its own unique set of arms that are not ideal. The authors overcome this by defining S_1, the set of all arms that included in at least one stepwise ideal set of arms. They define S_2 as the complement $[K]\backslash S_1$, the set of arms that is not ideal given the immediate known costs of any step.
Using this construct, the proof of *Theorem 6* merely adds the additional step of lower bounding regret caused by the stepwise non-ideal arms by the regret caused by arms in S_2. Aside from that, the proof is identical.
The implication of this is that the regret under CC-UCB2 is $o(\log T)$.

Online Policy Algorithm - Upper Bound
One of the assumptions made by the authors is that the cost of each arm is strictly positive. Let $Y_{i,t} \in [l_i, h_i]$ and denote $\underline{\Delta}_i := l_i - \theta_i$ the disparity between the lower range of the costs for arm i and its expected reward.
Theorem 5 upper bounds the cumulative regret under Algorithm 2 assuming that $\alpha \geq 1.5$ by:

$$R(T) \leq \sum_{i \in S_2} h_i \frac{4\alpha \log T}{\underline{\Delta}_i^2} + O(1)$$

The implication of this is that the regret under CC-UCB2 is $O(\log T)$ if S_2 is not empty. If $S_2 = \emptyset$ then the regret is bounded by a *constant.*
Similarly to earlier, regret can be decomposed into three parts.
Empirical Average Misestimation - Limiting the definition of misestimation, $\mathcal{E}_t$, to only the sample reward, *Lemma 1* and *Lemma 2* still hold. Therefore, the regret caused by misestimation is bounded by a constant, $K(1 + \frac{4\pi^2}{3})$.
Incorrectly ordering the ideal set of arms
To adapt the proof from unknown costs, we introduce several new definitions.

1. The authors partition the possible sets of instant costs to M partitions. These partitions are defined by the optimal lists of arms that they induce. Multiple known immediate costs are included in partition $\mathcal{Y}_m, m \in [M]$ corresponding to I_m, the ideal list of arms for that instant cost partition.

2. The likelihood for the instant costs to be part of partition m is $\rho_m := \mathcal{P}[Y_t \in \mathcal{Y}_m], \forall m \in [M]$.

3. The number of times an arm was pulled under partition m before step t is denoted $N_{i,t}^m$.

With these definitions, the authors upper bound the regret generated by incorrectly ordering the ideal set of arms. To do so, they partition the steps into M partitions depending on the ideal list of arms induced by the instant costs. They adapt *Lemma 10* so that it plays the same role in allowing the Hoeffding inequality to be applied.

The same logic process applied in *Lemma 3* is applied to each partition m, scaling the expected value of all steps using ρ_m. This slightly changes the exact terms used to bound the number of steps in a given partition with incorrect ordering and no misestimation.

The result is bounding each partition $\mathcal{Y}_m$ with a constant:

$$\mathbb{E}[\sum_{t=1}^{T} \mathbb{1}(\bar{\mathcal{E}}_t)\mathbb{1}(\mathcal{B}_t)\mathbb{1}(Y_t \in \mathcal{Y}_m)] \leq \rho_m(\sum_{j=2}^{|I^m|}(\zeta_j^m + \frac{2}{(\rho_m p_{j^*})^2}) + \xi_0) := \zeta^m$$

Pulling non-ideal arms - Lemma 7 upper bounds the regret caused by pulling non-ideal arms in a single step, replacing the bound offered by *Lemma 4* with one based on the known immediate costs.

$$\mathbb{E}[reg_t|\tilde{I}_t, Y_t] \leq \sum_{i \in \tilde{I} \setminus I_t^*} Y_{i,t}$$

The same logic is applied to bound the regret caused by pulling non-ideal arms in this setting as in the unknown cost setting with two primary changes.

1. Replacing unknown costs with known instant costs - This changes the proof by requiring three changes:
   a. *Lemma 7* replaces unknown costs with known instant costs.
   b. We upper bound the known instant costs using h_i.
   c. The bound developed in *Lemma 6* for the expected times each arm is pulled in S_2 can be reduced from $\frac{16\alpha \log T}{\Delta_i^2}$ to $\frac{4\alpha \log T}{\underline{\Delta}_i^2}$

2. Replacing the arms causing the regret - Instead of summing over the non-ideal arms, we sum over S_2

Regret caused by different cost partitions - Unlike in the unknown cost setting, CC-UCB2 incurs regret at step t when it pulls arms that are ideal under some cost partitions but not under Y_t.

To bound this regret, we define several new objects.

1. Minimal cost of arm i over all partitions, $l_i^m := min\{Y_{i,t}|Y_t \in \mathcal{Y}_m\}$
2. Minimal difference between the smallest cost of a given arm and its expected reward under partition $\mathcal{Y}_m$, $\tilde{\Delta}_i := \min_{m;i\notin I^m}(l_i - \theta_i)$. The proof only uses $\tilde{\Delta}_i$ that are positive because the regret is generated by arms that are ideal under some partition but not under m, $i \in S_1 \backslash I_m$. Therefore, $Y_{i,t} \geq l_i^m > \theta_i$.
3. We define $\hat{N}_{i,t}^m$ as the the number of times that arm i was pulled for time step t between $[\frac{t}{2}]$ and t under partition $\mathcal{Y}_m$.

We will outline the proof for this new type of regret not found in the previous setting:
1. ***Formulate the regret*** as $\mathbb{E}[\sum_{t=1}^T \mathbb{1}(\bar{\mathcal{E}}_t)\mathbb{1}(\bar{\mathcal{B}}_t)\sum_{i\in\tilde{I}_t\backslash(S_2\cup I_t^*)} Y_{i,t}]$
2. ***Sum over separate partitions of $\mathcal{Y}_m$***
3. ***Reformulate as bound on the number of steps arm i was pulled*** - We switch $\hat{c}_{i,t}$ with h_i and use the same process as *Lemma 3*.
4. ***Bound $\hat{N}_{i,t}^m$*** - Use *Lemma 8* (See Appendix) to bound $\hat{N}_{i,t}^m$ by exponentially diminishing functions of t referred to as $C_{i,t}^m$.
5. ***Bound the number of steps arm i was pulled*** - Use $\hat{m}$, the cost partition for which arm i was pulled most, to upper bound $C_{i,t}^m$ with $C_{i,t}^{\hat{m}}$. We also define n_i as a function of $\alpha, \rho_{\hat{m}}, p_i, \tilde{\Delta}_i^2$ s.t. $\frac{\rho_{\hat{m}} p_i t}{4} \geq \frac{4\alpha \log t}{\tilde{\Delta}_i^2}$. With that we bound the number of steps that arm i was pulled by $(n_i + \sum_{t=1}^\infty C_{i,t}^{\hat{m}})h_i$. This uses similar processes to the part of *Lemma 3* when we deal with the case of $\mathbb{1}(\tau_n \leq 2n)$.
6. ***Combining Results*** - We receive that this type of regret is bounded by a constant:

$$\mathbb{E}[\sum_{t=1}^T \mathbb{1}(\bar{\mathcal{E}}_t)\mathbb{1}(\bar{\mathcal{B}}_t) \sum_{i\in\tilde{I}_t\backslash(S_2\cup I_t^*)} Y_{i,t}] \leq \sum_{i\in S_1}(n_i + \sum_{t=1}^\infty C_{i,t}^{\hat{m}})h_i := \xi$$

Together, the regret generated UCB violation, incorrect ordering of ideal arms and different cost partitions are bounded by constants, contributing $O(1)$ to the cumulative regret. Pulling non-ideal arms contributes $O(\log T)$ regret. Therefore, the upper bound of Algorithm 2 is $O(\log T)$, like the lower bound. Thus, algorithm 2 has order-optimal regret performance.

2.4 Fail-Safe CCB

In this final section of the paper, the authors consider fining the algorithm for not achieving any reward that turn. This fine encourages pulling risky arms to ensure that reward is achieved each step. We refer to this fixed cost as C.

Optimal Offline Policy

In this case, the optimal policy is to order the arms in descending order of $\frac{\theta_i}{c_i}$ such that optimal arms uphold $\frac{1}{1+C} \leq \frac{\theta_i}{c_i}$.

The net reward in this setting is:

$$r_t := 1 - \prod_{i=1}^{|\tilde{I}_t|}(1 - X_{\tilde{I}_t(i),t}) - \sum_{i=1}^{|\tilde{I}_t|} Y_{\tilde{I}_t(i),t} - C\prod_{i=1}^{|\tilde{I}_t|}(1 - X_{\tilde{I}_t(i),t})$$

They prove this by showing that if we have I and $I' = [I, i]$ s.t. $i \in [K]\backslash I$, then $\mathbb{E}[r_t(I')] - \mathbb{E}[r_t(I)] > 0$ when $(1 + C)\theta_i > c_i$.

Online Policy Algorithm
Both CC-UCB and CC-UCB2 can be adapted with a I_t set by the threshold $\frac{1}{1+C}$, instead of 1. The upper and lower bounds are scaled slightly differently with the new threshold, but maintain $O(\log T)$ and $o(\log T)$ respectively. The proof for both unknown and instantly known regret bounds are identical to those outlined earlier.

3 Expansion - Best Arm Identification

The CC-UCB algorithm addresses a novel setting in the field of cascading bandits. Each arm is associated with reward and cost distributions. Often the cascading bandit setting assumes a set number of arms in the list played each round. However, CC-UCB must define a cutoff for "worth-while" arms because the cost may outweigh the benefit of playing an arm.
We wish to expand this setting to best-arm identification (BAI). CC-UCB seeks to minimize cumulative regret, thus it balances exploration and exploitation via the UCB bounds. The authors direct their algorithm to high-stake scenarios, such as dynamic treatment allocations, where exploitation must be pursued during exploration for ethical reasons. In contrast, BAI aims to discover the optimal arm as quickly as possible, ignoring the regret incurred during the exploration phase. This goal is often associated with tasks like advertising and recommendation systems, where a set budget may be assigned ahead of time to find the best ad or merchandise to be recommended. We suggest that cost can still play a part in tasks that allow for a pure exploration phase. For example, advertisers may want to use a cost-aware algorithm to mitigate risk associated with an ad or possibly offensive merchandise. Additionally, low-stakes medical setting like cosmetics or trials first performed on animals may also be relevant examples.

To do so, we suggest integrating into the CCB setting a modified concept from CascadeBAI [11]. CascadeBAI is an (ϵ, δ, k)-PAC algorithm. The algorithm receives L Bernoulli arms and outputs the k best arms with ϵ-error and $1 - \delta$ certainty. While CascadeBAI solves the cascading bandit problem, it is not appropriate for the setting of our reviewed paper. It does not offer an order over the optimal arms; it requires the user to decide up front the number of arms that are optimal and it does not take into consideration the cost of pulling each arm.

Therefore, we offer CC-BAI, Cost-aware Cascading Best-Arm Identification. We will show a number of adjustments made to adapt the BAI goal to CC-UCB's unique setting. We will also perform a simulation to compare them.
Instead of applying the algorithm over Bernoulli arms representing the reward, we implement a cost-aware Bernoulli variable. The authors find that arm i is ideal when $\theta_i > c_i$. Therefore, we suggest modeling the Bernoulli variable to be $\rho_i = \mathbb{1}\theta_i > c_i\}$. This maintains the insight of our paper for the conditions under which an arm should be pulled. However, instead of only approximating the expected reward and punishment like in CC-UCB, CC-BAI also models the probability that the the expected reward is greater than the expected punishment.
Next, we must replace the goal of discovering the k optimal arms. We suggest replacing the fixed number of K arms with a cutoff for the desired probability that the arm is part of the ideal list, $\mathcal{P}$. Therefore, instead of recovering the k optimal arms with ϵ-error and δ-certainty, we recover with ϵ-error and δ-certainty the arms that are likely to provide higher benefit than cost, $\mathcal{P}$ of the time. Thus the user will receive w.p. $1 - \delta$ a set of arms for whom the expected reward is higher than the cost, $\mathcal{P}$ of the time, with error ϵ. This set of arms can now be ordered according to the ratio of expected reward and expected cost, similar to the concept in CC-UCB
The original algorithm from [11], CascadeBAI and the modified algorithm we suggest, CC-BAI, appear in the Appendix.

We ran a simulation [1] to demonstrate experimental results comparing CC-BAI to CC-UCB. Because CC-BAI does not seek to optimize regret during the exploration phase, we compare the two algorithms during an exploitation phase.

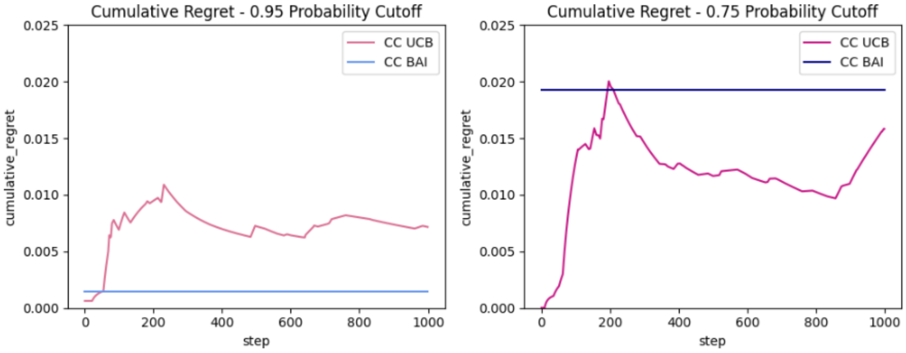

Figure 1: Exploitation Regret - Comparing CC-UCB and CC-BAI

For CC-BAI, the exploration phase runs until the algorithm converges. For

[1] `https://github.com/dparnas/cc_bai/blob/main/Bandit_Simulation.ipynb`

higher probability cutoffs, the exploration phase is longer than for lower probability cutoffs. After exploration, CC-BAI identifies a single ordered list of ideal arms to play each step during the exploitation phase. While the CC-UCB algorithm has no defined exploration phase, we defined its exploitation phase as the number of rounds that correspond to CC-BAI's exploitation phase. Therefore, if exploration took CC-BAI M steps, than the cumulative regret is compared only for steps $M + 1$ onward.

The results of the simulation indicate that CC-BAI may result in better regret when the probability cutoff is high. When we chose a 0.95 probability cutoff, the cumulative regret of CC-BAI was less than CC-UCB. However, for a probability cutoff of 0.75, CC-UCB beats the cumulative regret of CC-BAI most steps. These results make sense. If a high probability cutoff is chosen, the exploration phase will provide a more with higher probability a list of arms resulting in lower regret in the exploitation phase at the expense of prolonging the exploration phase. If a low probability cutoff is chosen, the exploration phase length will be reduced. However, the set of arms are more likely not to be ideal. Therefore, CC-UCB's continuous exploration-exploitation trade-off may be favorable.

Conclusion

We learned a new and interesting setting, cost-aware cascading bandits. We saw a UCB based algorthim using a confidence padding element both on the cost and on the reward, and presented the proof for it's lower and upper bounds in various settings. We suggested a new algorithm, CC-BAI, which offers a first step towards expanding the cost-aware cascading bandit setting to the best-arm identification task. In our initial simulation, we demonstrated that high probability cutoffs may offer improved regret in the exploitation phase over a UCB based continual exploration-exploitation. More rigorous theoretical analysis and further numerical testing would be necessary to suggest broader applications of CC-BAI.

References

[1] Chao Gan, Ruida Zhou, Jing Yang, and Cong Shen. Cost-aware cascading bandits. 68:3692–3706.

[2] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press.

[3] Shahin Shahrampour, Mohammad Noshad, and Vahid Tarokh. On sequential elimination algorithms for best-arm identification in multi-armed bandits. 65(16):4281–4292. Conference Name: IEEE Transactions on Signal Processing.

[4] Cong Shen. Universal best arm identification. 67(17):4464–4478. Conference Name: IEEE Transactions on Signal Processing.

[5] Long Tran-Thanh, Archie Chapman, Enrique Munoz de Cote, Alex Rogers, and Nicholas R. Jennings. Epsilon–first policies for budget–limited multi-armed bandits. 24(1):1211–1216.

[6] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas Jennings. Knapsack based optimal policies for budget–limited multi–armed bandits. 26(1):1134–1140.

[7] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. Multi-armed bandit with budget constraint and variable costs. 27(1):232–238.

[8] Yingce Xia, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, and Tao Qin. Budgeted bandit problems with continuous random costs. In *Asian Conference on Machine Learning*, pages 317–332. PMLR. ISSN: 1938-7228.

[9] Junpei Komiyama, Junya Honda, Hiroshi Nakagawa, and Komiyama Info. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays.

[10] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model.

[11] Zixin Zhong, Wang Chi Cheung, and Vincent Tan. Best arm identification for cascading bandits in the fixed confidence setting. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11481–11491. PMLR.

[12] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

4 Appendix

4.1 CC-UCB

The UCB padding is defined as $u_{i,t} := \sqrt{\frac{\alpha \log t}{N_{i,t}}}$

Algorithm 1 Cost-aware Cascading UCB (CC-UCB).

```
Input: $\epsilon, \alpha$;
Initialization: Pull all arms in $[K]$ once and observe their states and costs.
while t do
    for i=1 : K do
        $U_{i,t} = \hat{\theta}_{i,t} + u_{i,t}$;
        $L_{i,t} = \max(\hat{c}_{i,t} - u_{i,t}, \epsilon)$; For CC-UCB2, $Y_{i,t}$ replaces $\hat{c}_{i,t}$
        if $U_{i,t}/L_{i,t} > 1$ then $I_t \leftarrow i$;
        end if
    end for
    Rank arms in $I_t$ in descending order based on $\frac{U_{i,t}}{L_{i,t}}$.
    for $i = 1 : |I_t|$ do
        Pull arm $I_t(i)$ and observe $X_{I_t(i),t}, Y_{I_t(i),t}$
        $\tilde{I}_t \leftarrow i$;
        if $X_{I_t(i),t} = 1$ then break
        end if
    end for
    Update $N_{i,t}, \hat{\theta}_{i,t}, \hat{c}_{i,t}$ for all $i \in \hat{I}_t$
    t = t+1;
end while
```

4.2 Incorrectly ordering the ideal set of arms

”Divide and Conquer”
We iterate only over steps with no misestimation, we bound the above term in two scenarios - when most steps do not have misestimation up to τ_n and when most steps do, $\mathbb{1}(\tau_n \leq 2n)$ or $\mathbb{1}(\tau_n > 2n)$. We do so by bounding the term, $\frac{\alpha \log \tau_n}{\Delta^2_{(j-1)^*,j^*}}$.

1. Most steps do not have misestimation for τ_n - $\mathbb{1}(\tau_n \leq 2n)$ - Using the bound on τ_n imposed by $\mathbb{1}(\tau_n \leq 2n)$, we bound $\frac{\alpha \log 2n}{\Delta^2_{(j-1)^*,j^*}} \leq \frac{p_{j^*}}{2} n$ for any $n >= \zeta_j$ s.t. ζ_j is a constant depending on the α, p^*_j and $\Delta^2_{(j-1)^*,j^*}$. Using **lemma 10** (see appendix), we reformulate the number of times that each ideal arm is pulled as a sum of bernoulli variables. Together, these allow us to use Hoeffding to bound the expected number of times that each arm is pulled at each step without misestimation by $\zeta_j + \frac{2}{p^2_{j^*}}$.

2. Most steps have misestimation for τ_n - $\mathbb{1}(\tau_n > 2n)$ - We reformulate the bound to limit the expected number of steps without misestimation, which should be small by design. This is equivalent to the probability of having many steps with misestimation. Using Chebyshev's inequality and by bounding the expectation of having misestimation, we find a bound that depends on the number of arms and the step t. We bound the sum of the ensuing series by a constant $\xi_0 := 16K(\frac{\pi^2}{6} + 1 + \log 2 + \frac{1}{3}(2 + \log^2 3 + 2\log 3))$.

4.3 Lemma 4

Lemma 4 states that given an ordered list I_t that includes all arms from I^* with the ideal ordering, the regret under Algorithm 1 is as follows:

$$\mathbb{E}[reg_t|\tilde{I}_t] \leq \sum_{i \in \tilde{I}_t \setminus I^*} c_i$$

1. Like in the lower bound, we consider $\tilde{i}_t$, the last non-ideal arm pulled. If that arm did not realize a reward, $X_{\tilde{i},t} = 0$, then the regret comes from the cost of pulling all of the non-ideal arms, $\tilde{I}_t \setminus I^*$. Thus, the regret is bounded by the cost of those non-ideal arms.

$$\mathbb{E}[r_t^* - r_t|\tilde{I}_t, X_{\tilde{i},t} = 0] = \sum_{i \in \tilde{I}_t \setminus I^*} c_i$$

2. Otherwise, if the last non-ideal arm did realize a reward, $X_{\tilde{i},t}$ then the regret is the reward from the ideal arms without the reward and costs of the non-ideal arms pulled. Because the largest reward from the ideal arms would be 1 and the non-ideal arm realized a reward of 1, these cancel out in the upper bound. This leaves the cost of the non ideal arms. That is to say $\sum_{i \in \tilde{I}_t \setminus I^*} c_i$.
Combining both cases, we conclude that the regret caused by pulling non-ideal arms is bounded by $\sum_{i \in \tilde{I}_t \setminus I^*} c_i$ as **Lemma 4** states.

4.4 Lemma 10

This technical lemma aids us to bound the steps with incorrectly ordered ideal arms. **Lemma 10** lower bounds the number of times that arm $i \in I^*$ is chosen. It proves that the number of times arm i is chosen is at least the number of steps that there is no misestimation in any of the arms and all of the arms aside from i have a state of 0. This is because arm i will only be pulled if the arms preceding it in $\tilde{I}_t$ have a state of 0. All the more so, if all the states of the other arms are 0. The authors designate random variable $Z_{i,t}$ as the event of this happening as follows:

$$Z_{i,t} = \begin{cases} 0, & \text{if } \bar{\mathcal{E}}_{\mathfrak{t}} = 0 \\ 0, & \text{if } \bar{\mathcal{E}}_{\mathfrak{t}} = 1 \text{ and } \exists j \in [K]\setminus\{i\}, X_{j,t} = 1 \\ 1, & \text{if } \bar{\mathcal{E}}_{\mathfrak{t}} = 1 \text{ and } \forall j \in [K]\setminus\{i\}, X_{j,t} = 0 \end{cases} \tag{1}$$

This can be rewritten with $p_i := \frac{\prod_{j=1}^{K}(1-\theta_j)}{(1-\theta_i)}$ as:

$$Z_{i,t} = \begin{cases} 0, & \text{if } \bar{\mathcal{E}}_{\mathfrak{t}} = 0 \\ Ber(p_i), & \text{if } \bar{\mathcal{E}}_{\mathfrak{t}} = 1 \end{cases} \tag{2}$$

This lemma was used to bound the case of $\mathbb{1}(\tau_n \leq 2n)$.

4.5 Lemma 8

This technical lemma helps to bound the number of times an arm is pulled under partition $\mathcal{Y}_m$ as defined above.
Lemma 8 bounds $\forall i \in S$, any m s.t. $i \in I^m, t > 2k$:

$$\mathcal{P}[\hat{N}_{i,t}^m \leq \frac{\rho_m p_i t}{t}] \leq 2k(\frac{t}{2}+1)(\frac{t}{2})^{-2\alpha+1} + exp(\frac{-\rho_m^2 p_i^2 t}{16}) := C_{i,t}^m$$

The proof splits the $\mathcal{P}[\hat{N}_{i,t}^m]$ into two scenarios - when there is misestimation in one of the steps between $\frac{t}{2}$ and t when there is not.
1. The total probability of there being misestimation in one of the steps between $\frac{t}{2}$ and t can be bounded by bounding each arms likelihood to be misestimated. Hoeffding is used to bound the probability of this happening for any of the arms. This case is bounded by $2k(\frac{t}{2}+1)(\frac{t}{2})^{-2\alpha+1}$.
2. If there is no misestimation in any step, we will adapt the logic of *Lemma 10* for $\hat{Z}_{i,t}^m \sim Ber(\rho_m p_i)$, an indicator that under cost partition m arms $[K]\backslash\{i\}$ were not pulled. Similarly to *Lemma 10*, $\hat{N}_{i,t}^m \geq \sum_{n=[\frac{t}{2}]}^{t} \hat{Z}_{i,n}^m$. This sum of random variables is bounded using hoeffding and we get an upper bound of $exp(\frac{-\rho_m^2 p_i^2 t}{16})$.

4.6 CascadeBAI Algorithm

Algorithm 2 Cost-aware Cascading Best-Arm Identification (CC-BAI).

```
**Input**: risk $\delta$, tolerance $\epsilon$, size of arm K;
**Initialization**: $t = 1, D_1 = [L], A_1 = \emptyset, R_1 = \emptyset, T_0(i) = 0, \hat{w}_0(i) = 0, \forall i$
**while** $D_t \neq \emptyset$, $|A_t| < K$ and $|R_t| < L - K$ do **do**
    Sort the items in $D_t$ according to the number of previous observations: $T_{t-1}(i_1^t) \leq ... \leq T_{t-1}(i_{|D_t|}^t)$
    **if** $|D_t| \geq K$ then **then**
        pull arm $S_t = (i_1^t, i_2^t, ...i_1^K)$.
    **else**
        pull arm $S_t = (i_1^t, i_2^t, ...i_{|D_t|}, S_t')$, where $S_t'$ is any $(K - |D_t|)$- subset of $A_t \cup R_t$
    **end if**
    Observe $\tilde{k_t} \in \{1, ...., K, \infty\}$
    for each $i \in D_t$, if $W_t(i)$ is observed then
        $$\hat{w}_t(i) = \frac{T_{t-1}(i)\hat{w}_{t-1}(i) + W_t(i)}{T_{t-1(i)+1}}, T_t(i) = T_{t-1}(i) + 1.$$
    Otherwise, $\hat{w}_t(i) = \hat{w}_{t-1}(i), T_t(i) = T_{t-1}(i)$.
    $k_t = K - |A_t|$
    Calculate the UCBs and LCBs for each $i \in D_t$:
        $$U_t(i, \delta) = \hat{w}_t(i) + C_t(i, \delta),$$
        $$L_t(i, \delta) = \hat{w}_t(i) - C_t(i, \delta).$$
    Find items in $D_t$ that have the $k_t^{th}$ and the $(k_t + 1)^{st}$ largest empirical means:
    $j' = \arg\max^{(k_t)}_{j \in D_t} \hat{w}_t(j)$,
    $j* = \arg\max^{(k_t+1)}_{j \in D_t} \hat{w}_t(j)$,
    $A_{t+1} = A_t \cup \{i \in D_t | L_t(i, \delta) > U_t(j', \delta) - \epsilon\}$.
    $R_{t+1} = R_t \cup \{i \in D_t | L_t(i, \delta) < U_t(j*, \delta) - \epsilon\}$.
    $D_{t+1} = D_t / (R_{t+1} \cup A_{t+1})$.
    $t = t + 1$
    If $|A_t| = K$, output $A_t$; otherwise, output the first $K$ items that entered $A_t$
**end while**
```

4.7 CC-BAI

Algorithm 3 Cost-aware Cascading Best-Arm Identification (CC-BAI).

```
Input: risk $\delta$, tolerance $\epsilon$, probability cutoff $\mathcal{P}$;
Initialization: $t = 1, D_1 = [L], A_1 = \emptyset, R_1 = \emptyset, T_0(i) = 0, \hat{w}_0(i) = 0, \forall i$
while $D_t \neq \emptyset$ do do
    Sort the items in $D_t$ according to the number of previous observations: $T_{t-1}(i_1^t) \leq ... \leq T_{t-1}(i_{|D_t|}^t)$
    pull arms $S_t = (i_1^t, i_2^t, ...i_{|D_t|}^t)$
    Observe $X_{i_j^t}, c_{i_j^t}, \forall j \in \{1, ...., |D_t|\}$ until reward achieved and update the averages $\bar{X}_{i_j^t}, \bar{c}_{i_j^t}$
    $\forall j \in \{1, ...., |D_t|\}$ that was observed,
        $$\hat{w}_t(i) = \frac{T_{t-1}(i)\hat{w}_{t-1}(i) + \mathbb{I}(\bar{X}_{i_j^t} > \bar{c}_{i_j^t})}{T_{t-1(i)} + 1}, T_t(i) = T_{t-1}(i) + 1.$$
    Otherwise, $\hat{w}_t(i) = \hat{w}_{t-1}(i), T_t(i) = T_{t-1}(i)$.
    Calculate the UCBs and LCBs for each $i \in D_t$:
        $$U_t(i, \delta) = \hat{w}_t(i) + C_t(i, \delta),$$
        $$L_t(i, \delta) = \hat{w}_t(i) - C_t(i, \delta).$$
    Find items in $D_t$ whose empirical means are immediately above and below $\mathcal{P}$, $(\mathcal{P}^+)$ and $(\mathcal{P}^-)$ respectfully:
    $j^+ = \arg\max_{j \in D_t}^{(\mathcal{P}^+)} \hat{w}_t(j)$,
    $j^- = \arg\max_{j \in D_t}^{(\mathcal{P}^-)} \hat{w}_t(j)$,
    $A_{t+1} = A_t \cup \{i \in D_t | L_t(i, \delta) > U_t(j^-, \delta) - \epsilon\}$.
    $R_{t+1} = R_t \cup \{i \in D_t | U_t(i, \delta) < L_t(j^+, \delta) - \epsilon\}$.
    $D_{t+1} = D_t / (R_{t+1} \cup A_{t+1})$.
    $t = t + 1$
end while
Order $A_t$ by $\frac{\bar{X}_i}{\bar{c}_i}$ and output ordered $A_t$
```