Dynamic Planning and Learning under Recovering Rewards — Supplementary Material

David Simchi-Levi [1] **Zeyu Zheng** [2] **Feng Zhu** [1]

1. Proof of NP-Hardness of the offline problem

Here we adopt the similar idea in (Cella & Cesa-Bianchi, 2020) to give a proof of NP-Hardness of the offline problem, even if we confine ourselves to find a long-run optimal policy within the class of purely periodic policies.

Proof.

Our proof relies on a reduction from the Periodic Maintenance Scheduling Problem (PMSP) to our problem. In PMSP, we are given n machines for service, and n positive integers $\ell_1, \cdots, \ell_n$ such that $\sum_{i=1}^{n} 1/\ell_i \leq 1$. We call $\{\ell_i\}_{i\in[N]}$ is feasible, if there exists a schedule such that the consecutive service times of each machine i are exactly ℓ_i times apart, and meanwhile in each time period at most 1 machine is in service. The question is to examine whether $\{\ell_i\}_{i\in[N]}$ is feasible. Bar-Noy et al. (2002) showed that PMSP is NP-complete.

Given an instance of PMSP with $\ell_1, \cdots, \ell_n$, we prove that $\{\ell_i\}_{i\in[N]}$ is feasible if and only if there exists a 1-PPP such that its long-run average reward is $\sum_{i\in[N]} 1/\ell_i$. We let $N = n$, $K = 1$, and

$$R_i(d) = \begin{cases} 0, & \text{if } d < \ell_i, \\ 1, & \text{if } d \geq \ell_i. \end{cases}$$

On one hand, if this instance of PMSP is feasible, then we can directly apply the corresponding schedule to pull arms, yielding a long-run average reward

$$\sum_{i\in[N]} R_i(\ell_i)/\ell_i = \sum_{i\in[N]} 1/\ell_i.$$

Moreover, this schedule is purely periodic.

On the other hand, suppose we can find a purely periodic schedule of arms pulling such that the long-run average reward is no less than $\sum_{i\in[N]} 1/\ell_i$. Note that the long-run average reward of pulling an arm i is upper bounded by

$$R_i(d)/d \leq 1/\ell_i \ (\forall d \geq 1),$$

and the equality holds iff $d = \ell_i$. Therefore, we must pull arm i every ℓ_i times eventually, or the average reward within one period is strictly less than $1/\ell_i$. This means that the instance is feasible.

2. Proofs of Lemmas and Theorems

Proof of Lemma 1.

[1]Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Massachusetts, USA [2]Department of Industrial Engineering and Operations Research, University of California, Berkeley, USA. Correspondence to: David Simchi-Levi <dslevi@mit.edu>, Zeyu Zheng <zyzheng@berkeley.edu>, Feng Zhu <fengzhu@mit.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Let $F^{\text{cc}}_{i,T}(x)$ be the value of the following problem.

$$\begin{aligned} \max_{s} \quad & \frac{1}{T}\sum_{j=1}^{J} R^{\text{cc}}_i(s_j - s_{j-1}) \\ \text{s.t.} \quad & J \le x \cdot T, \\ & 0 = s_0 < s_1 < \cdots < s_J \le T. \end{aligned} \tag{1}$$

Intuitively, $F^{\text{cc}}_{i,T}(x)$ is the optimal average reward of i under $\{R^{\text{cc}}_i(d)\}$, given that we pull arm i no more than $x \cdot T$ times. Then we can see that $F^{\text{cc}}_{i,T}(x) \ge F_{i,T}(x)$ $(\forall i \in [N], T \ge 1, x \in [0,1])$.

Claim 1. Fix $i \in [N]$, $T \ge 1$ and $x \in [0,1]$, then there exists an optimal solution to (1) such that $s_J = T$ and $\{s_j - s_{j-1}\}$ can take at most two different values.

Apparently, under Assumption 1, the objective value will never decrease if we let $s_J = T$. Thus, we can always assume $s_J = T$. For simplicity, we write $ds_j \triangleq s_j - s_{j-1}$. For any feasible solution $\{s_j\}$ and J of (1), we define

$$\text{dist}(\{s_j\}_{j=1}^{J}) = \sum_{1 \le i,j \le J} |ds_j - ds_i| \cdot \mathbb{1}\{|ds_j - ds_i| > 1\}.$$

Let $\{s^*_j\}_{j=1}^{J^*}$ be an optimal solution such that dist $\left(\{s^*_j\}_{j=1}^{J^*}\right)$ attains the minimum among all optimal solutions. This is attainable since the number of feasible solutions to (1) is finite, and as a result, there always exists an optimal solution and the number of optimal solutions is finite. Suppose dist $\left(\{s^*_j\}_{j=1}^{J^*}\right) > 0$, then we choose $j_1 = \arg\min_j \left\{ds^*_j\right\}$ and $j_2 = \arg\max_j \left\{ds^*_j\right\}$. Without loss of generality, we assume $j_1 < j_2$. Then $ds^*_{j_2} - ds^*_{j_1} \ge 2$. We define a new set $\{s'_j\}$ as follows.

$$s'_j = \begin{cases} s^*_j + 1, & \text{if } j_1 \le j < j_2, \\ s^*_j, & \text{else.} \end{cases}$$

Note that $s'_J = s^*_J = T$. Then

$$\begin{aligned} ds'_{j_1} &= ds^*_{j_1} + 1 \le ds^*_{j_2} - 1 = ds'_{j_2}, \\ ds'_j &= ds^*_j, \quad \forall j \ne j_1, j_2. \end{aligned}$$

By our choice of j_1 and j_2 ,we have

$$\begin{aligned} \left|ds'_{j_k} - ds'_j\right| \mathbb{1}\left\{\left|ds'_{j_k} - ds'_j\right| > 1\right\} &\le \left|ds^*_{j_k} - ds^*_j\right| \mathbb{1}\left\{\left|ds^*_{j_k} - ds^*_j\right| > 1\right\}, \quad \forall k \in \{1,2\}, \\ \left|ds'_{j_2} - ds'_{j_1}\right| \mathbb{1}\left\{\left|ds'_{j_2} - ds'_{j_1}\right| > 1\right\} &< \left|ds^*_{j_2} - ds^*_{j_1}\right| = \left|ds^*_{j_2} - ds^*_{j_1}\right| \mathbb{1}\left\{\left|ds^*_{j_2} - ds^*_{j_1}\right| > 1\right\}. \end{aligned}$$

Thus dist $\left(\{s'_j\}_{j=1}^{J}\right) <$ dist $\left(\{s^*_j\}_{j=1}^{J}\right)$. However, since $\{R^{\text{cc}}_i(d)\}$ is concave, we have

$$\begin{aligned} & R^{\text{cc}}_i(ds'_{j_1}) + R^{\text{cc}}_i(ds'_{j_2}) - R^{\text{cc}}_i(ds^*_{j_1}) - R^{\text{cc}}_i(ds^*_{j_2}) \\ = & \left(R^{\text{cc}}_i(ds'_{j_1}) - R^{\text{cc}}_i(ds'_{j_1} - 1)\right) - \left(R^{\text{cc}}_i(ds^*_{j_2}) - R^{\text{cc}}_i(ds^*_{j_2} - 1)\right) \ge 0. \end{aligned}$$

This means either $\{s^*_j\}_{j=1}^{J^*}$ is not optimal, or it does not have the minimum dist value. A contradiction. Therefore, dist $\left(\{s^*_j\}_{j=1}^{J^*}\right) = 0$, indicating $\{ds^*_j\}$ must take at most two different values.

Claim 2. Let $x = \alpha\frac{1}{d+1} + (1-\alpha)\frac{1}{d}$, where $d \in \mathbb{Z}_+$, $d \ge d^{(1)}_i$, and $\alpha \in (0,1]$. Then $F^{\text{cc}}_{i,T}(x) \le \alpha\frac{R^{\text{cc}}_i(d+1)}{d+1} + (1-\alpha)\frac{R^{\text{cc}}_i(d)}{d} = F_i(x)$.

From Claim 1, $\exists d' \in \mathbb{Z}_+$, an optimal scheduling of (1) satisfies $s_j - s_{j-1} \in \{d', d'+1\}$. Then we have

$$T \le (d'+1)J \le (d'+1)xT < (d'+1)T/d,$$

which indicates $d' \ge d$. Suppose in the optimal scheduling,

$$\begin{aligned} a &= \#\{j \in [J] : s_j - s_{j-1} = d'\}, \\ b &= \#\{j \in [J] : s_j - s_{j-1} = d'+1\}. \end{aligned}$$

Then

$$a+b \le x \cdot T, \quad ad' + b(d'+1) = T.$$

Since $\{R_i^{\text{cc}}(d)\}_{d\ge 0}$ is concave, we have $R_i^{\text{cc}}(d)/d$ is non-increasing. If $d' \ge d+1$, then we have

$$\begin{aligned} F_{i,T}(x) &= \frac{a}{T} R_i^{\text{cc}}(d') + \frac{b}{T} R_i^{\text{cc}}(d'+1) = \frac{ad'}{T}\frac{R_i^{\text{cc}}(d')}{d'} + \frac{b(d'+1)}{T}\frac{R_i^{\text{cc}}(d'+1)}{d'+1} \\ &\le \frac{R_i^{\text{cc}}(d+1)}{d+1} \le \alpha \frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d}. \end{aligned}$$

If $d' = d$, then

$$T = ad + b(d+1) \le (d+1)xT - a,$$

which means $\frac{ad}{T} \le 1-\alpha$. We thus have

$$\begin{aligned} F_{i,T}(x) &= \frac{a}{T} R_i^{\text{cc}}(d) + \frac{b}{T} R_i^{\text{cc}}(d+1) = \frac{ad}{T}\frac{R_i^{\text{cc}}(d)}{d} + \frac{b(d+1)}{T}\frac{R_i^{\text{cc}}(d+1)}{d+1} \\ &= \frac{ad}{T}\frac{R_i^{\text{cc}}(d)}{d} + \left(1 - \frac{ad}{T}\right)\frac{R_i^{\text{cc}}(d+1)}{d+1} \le \alpha \frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d}, \end{aligned}$$

since $\frac{R_i^{\text{cc}}(d)}{d} \ge \frac{R_i^{\text{cc}}(d+1)}{d+1}$.

We are left to show that $\alpha\frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d} = F_i(x)$, which means $\alpha\frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d}$ can be achieved in an asymptotic sense. Let $k \ge 1$ such that $d_i^{(k)} \le d < d+1 \le d_i^{(k+1)}$. Then

$$\begin{aligned} &\alpha\frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d} \\ &= \alpha\frac{R_i^{\text{cc}}(d_i^{(k)})\frac{d_i^{(k+1)}-(d+1)}{d_i^{(k+1)}-d_i^{(k)}} + R_i^{\text{cc}}(d_i^{(k+1)})\frac{(d+1)-d_i^{(k)}}{d_i^{(k+1)}-d_i^{(k)}}}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d_i^{(k)})\frac{d_i^{(k+1)}-d}{d_i^{(k+1)}-d_i^{(k)}} + R_i^{\text{cc}}(d_i^{(k+1)})\frac{d-d_i^{(k)}}{d_i^{(k+1)}-d_i^{(k)}}}{d} \\ &= R_i^{\text{cc}}(d_i^{(k)})\frac{xd_i^{(k+1)}-1}{d_i^{(k+1)}-d_i^{(k)}} + R_i^{\text{cc}}(d_i^{(k+1)})\frac{1-xd_i^{(k)}}{d_i^{(k+1)}-d_i^{(k)}} \\ &= R_i(d_i^{(k)})\frac{xd_i^{(k+1)}-1}{d_i^{(k+1)}-d_i^{(k)}} + R_i(d_i^{(k+1)})\frac{1-xd_i^{(k)}}{d_i^{(k+1)}-d_i^{(k)}} \end{aligned}$$

We solve

$$a+b = x \cdot T, \quad ad_i^{(k)} + bd_i^{(k+1)} = T$$

and get $a = \frac{d_i^{(k+1)}xT - T}{d_i^{(k+1)}-d_i^{(k)}}$ and $b = \frac{T - d_i^{(k)}xT}{d_i^{(k+1)}-d_i^{(k)}}$. Then

$$\lfloor a \rfloor + \lfloor b \rfloor \le x \cdot T, \quad \lfloor a \rfloor d_i^{(k)} + \lfloor b \rfloor d_i^{(k+1)} \le T.$$

We have

$$\begin{aligned} \liminf_T F_{i,T}(x) &\ge \lim_T \frac{\lfloor a \rfloor R_i(d_i^{(k)})}{T} + \frac{\lfloor b \rfloor R_i(d_i^{(k+1)})}{T} \\ &= R_i(d_i^{(k)})\frac{xd_i^{(k+1)}-1}{d_i^{(k+1)}-d_i^{(k)}} + R_i(d_i^{(k+1)})\frac{1-xd_i^{(k)}}{d_i^{(k+1)}-d_i^{(k)}} \\ &= \alpha\frac{R_i^{\text{cc}}(d+1)}{d+1} + (1-\alpha)\frac{R_i^{\text{cc}}(d)}{d}. \end{aligned}$$

Claim 3. Let $x \geq \frac{1}{d_i^{(1)}}$. Then $F_{i,T}(x) \leq \frac{R_i(d_i^{(1)})}{d_i^{(1)}} = F_i(x)$.

We can see from the definition of $d_i^{(1)}$ that

$$F_{i,T}(x) \leq \frac{1}{T}\sum_{j=1}^{J} R_i(s_j - s_{j-1}) = \frac{1}{T}\sum_{j=1}^{J} \frac{R_i(s_j - s_{j-1})}{s_j - s_{j-1}}(s_j - s_{j-1}) \leq \frac{R_i(d_i^{(1)})}{d_i^{(1)}}.$$

On the other hand, if we let $s_j = j \cdot d_i^{(1)}$, then

$$\liminf_T F_{i,T}(x) \geq \lim_T \frac{\lfloor T/d_i^{(1)} \rfloor}{T} R_i(d_i^{(1)}) = \frac{R_i(d_i^{(1)})}{d_i^{(1)}}.$$

□

Proof of Lemma 2.

We first prove the claim that (2) is an upper bound on the original problem. For any schedule within a finite time horizon T, let

$$x_i = \#\{t \in [T] : i \text{ is pulled at time period } t\}/T \in [0, 1].$$

Then $\sum_{i\in[N]} x_i \leq K$ always hold. Further, the reward collected from arm i is no less than $F_{i,T}(x_i) \leq F_i(x_i)$, by our definition of $F_{i,T}(x_i)$ and Lemma 1. Thus, the total reward collected is no less than

$$\sum_{i\in[N]} F_i(x_i).$$

Now we prove the remaining part. Suppose $\{x_i^*\}$ is a feasible solution of (2). Then making $x_i^* \leftarrow \min\{x_i^*, 1/d_{i,1}\}$ does not decrease the objective value. If more than one components are not of the form $\{1/d_i^{(k)}\} \cup \{0\}$, then we can assume $x_{i_1}^* \in (1/(d_{i_1,j_1+1}), 1/d_{i_1,j_1})$ and $x_{i_2}^* \in (1/(d_{i_2,j_2+1}), 1/d_{i_2,j_2})$, where $i_1 \neq i_2$. From Lemma 1, F_{i_k} $(k \in \{1, 2\})$ is linear on $[1/(d_{i_k,j_k+1}), 1/d_{i_k,j_k}]$, so we can move x_{i_1} larger (smaller) and x_{i_2} smaller (larger) by the same distance until one of them reach an endpoint. The objective value will not decrease for at least one direction, and meanwhile this will not violate the hard constraint, but strictly decrease the number of $i \in [N]$ that $x_i^* \notin \{1/d_i^{(k)}\} \cup \{0\}$ in the feasible solution. We can thus repeat the procedure above until the solutions is transformed into the property stated in Lemma 2. In fact, the procedure takes at most $\mathcal{O}(N)$ time to transform any feasible solution into the form we want.

□

Proof of Lemma 3.

When $1/x_i^* \in \{d_i^{(k)}\}_{k\geq 1}$, we have

$$\frac{R_i(d_i)/d_i}{F_i(x_i^*)} = \frac{R_i(d_i)/d_i}{R_i^{\text{cc}}(1/x_i^*)x_i^*} = \frac{R_i(d_i)/d_i}{R_i(1/x_i^*)x_i^*} \geq \frac{1}{d_i x_i^*},$$

where the first equality holds from Lemma 1, and the inequality holds from Assumption 1. We notice that

$$\{1, \cdots, a-1\} \bigcup \left\{a \times 2^\ell, (a+1) \times 2^\ell, \cdots, (2a-1) \times 2^\ell\right\}_{\ell\geq 0} \subset \mathcal{D}[a]$$

because any positive integer number no less than $2a$ can be written as a positive odd number (less than $2a$) times a power of 2. Now if $1/x_i^* \leq 2a - 1$, then $d = 1/x_i^*$. If $1/x_i^* \geq 2a$, then

$$d_i \leq \sup_{a\leq b<2a} \frac{b+1}{b} \cdot 1/x_i^* \leq \frac{a+1}{a x_i^*}.$$

Thus, $\frac{1}{d_i x_i^*} \geq \frac{a}{a+1}$.

□

Proof of Lemma 4.

Part 1. Without loss of generality, we assume that the sum of frequencies is strictly larger than 1 (otherwise we simply choose $\mathcal{I}_{j_1} = \mathcal{I}_j$). For each $1 \le k \le |\mathcal{I}_j|$, we write $d_{i_k} = (2j-1) \times 2^{\ell_{i_k}}$. Since the sum of frequencies is no less than 1, there exists some $1 \le m < |\mathcal{I}_j|$ such that

$$\sum_{k=1}^{m} 1/d_{i_k} \le 1 < \sum_{k=1}^{m+1} 1/d_{i_k}.$$

We will prove in the following that

$$\sum_{k=1}^{m} 1/d_{i_k} = 1.$$

In fact, we have

$$d_{i_m} - 1 \le d_{i_m} - d_{i_m}/d_{i_{m+1}} < \sum_{k=1}^{m} d_{i_m}/d_{i_k} = \sum_{k=1}^{m} 2^{\ell_{i_m} - \ell_{i_k}} \in \mathbb{Z}_+,$$

which indicates that

$$d_{i_m} \le \sum_{k=1}^{m} d_{i_m}/d_{i_k}.$$

This is what we desire. Apparently, we can find m by adding d_{i_k} one by one and compare each sum with 1. Moreover, once the sum reaches 1 (our proof above guarantees this), we can make the former m products into a group, and restart from product i_{m+1}. The total time complexity is $\mathcal{O}(|\mathcal{I}_j|)$.

Part 2. We begin with $j = 1$. We write $d_{i_k} = 2^{\ell_{i_k}}$ and let $\ell = \max_{i \in \mathcal{I}_1} \ell_i$. We use induction method to prove that after sorting, we can specify a 1-PPP in $\mathcal{O}(|\mathcal{I}_1| \log d_{i_{|\mathcal{I}_1|}})$ time. When $\ell = 0$, there is only one product in $\mathcal{I}_1$ with frequency 1. Let $t_{i_1} = 0$. The result is correct.

Suppose for ℓ the result is correct. Now consider the case for $\ell + 1$. Then $\ell_{i_1} \ge 1$ and

$$\sum_{i \in \mathcal{I}_1} 1/(d_i/2) \le 2.$$

If $\sum_{i \in \mathcal{I}_1} 1/(d_i/2) \le 1$, then by induction, we can specify a 1-PPP in $\mathcal{O}\left(|\mathcal{I}_1| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right)$ time. We project the offering time by $t \to 2t$. The total time complexity is

$$\mathcal{O}\left(|\mathcal{I}_1| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right) + \mathcal{O}\left(|\mathcal{I}_1|\right) = \mathcal{O}\left(|\mathcal{I}_1| \log d_{i_{|\mathcal{I}_1|}}\right).$$

If $\sum_{i \in \mathcal{I}_1} 1/(d_i/2) > 1$, then applying the proof of Part 1, we can split $\mathcal{I}_1$ into two parts $\mathcal{I}_{11}$ and $\mathcal{I}_{12}$ in $\mathcal{O}(|\mathcal{I}_1|)$ time such that

$$\sum_{i \in \mathcal{I}_{1k}} 1/(d_i/2) \le 1, \quad \forall k \in \{1, 2\}.$$

By induction, we can specify a feasible 1-PPP for $\mathcal{I}_{11}$ and $\mathcal{I}_{12}$. The time complexity for this procedure is

$$\mathcal{O}\left(|\mathcal{I}_{11}| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right) + \mathcal{O}\left(|\mathcal{I}_{12}| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right) = \mathcal{O}\left(|\mathcal{I}_1| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right).$$

Now we project the offering time by $t \to 2t - 1$ for products in $\mathcal{I}_{11}$ and $t \to 2t$ for products in $\mathcal{I}_{12}$. This fulfills our requirement. The total time complexity is

$$\mathcal{O}\left(|\mathcal{I}_1| \log\left(d_{i_{|\mathcal{I}_1|}}/2\right)\right) + \mathcal{O}\left(|\mathcal{I}_1|\right) = \mathcal{O}\left(|\mathcal{I}_1| \log d_{i_{|\mathcal{I}_1|}}\right).$$

We continue on general cases. When $j > 1$, we notice that

$$\sum_{i \in \mathcal{I}_j} 1/\left(d_i/(2j-1)\right) \le 2j-1.$$

By Part 1, we can split $\mathcal{I}_j$ into at most $2j-1$ disjoint sets such that $\mathcal{I}_j = \bigcup_s \mathcal{I}_{j_s}$ such that

$$\sum_{i \in \mathcal{I}_{j_s}} 1/\left(d_i/(2j-1)\right) \le 1, \quad \forall s.$$

By our proof for $j = 1$ above, we can specify a feasible 1-PPP for $\mathcal{I}_{j_s}$ $(\forall s)$. We project the offering time by $t \to (2j-1)(t-1)+s$ for products in $\mathcal{I}_{j_s}$. This completes the construction. The total time complexity is

$$\mathcal{O}(|\mathcal{I}_j|) + \sum_s \mathcal{O}\left(|\mathcal{I}_{j_s}| \log\left(d_{i_{|\mathcal{I}_j|}}/(2j-1)\right)\right) = \mathcal{O}\left(|\mathcal{I}_j| \log d_{i_{|\mathcal{I}_j|}}\right).$$

□

Proof of Theorem 1.

For each selected product $i \in [N]$, it is offered at time $t_i + kd_i$ $(k \ge 1)$ until T. Thus, the number of time it is offered is lower bounded by

$$\lfloor (T-t_i)/d_i \rfloor \ge \lfloor T/d_i \rfloor > T/d_i - 1.$$

The total reward of i throughout the whole time horizon is lower bounded by

$$(T/d_i - 2) \cdot R(d_i) = R_i(d_i)/d_i \cdot T - 2 \cdot R(d_i).$$

Therefore, the total reward obtained is lower bounded by

$$\begin{aligned}
&\sum_{i \text{ is selected}} F_i(1/d_i) \cdot T - 2 \sum_{i \text{ is selected}} R(d_i) \\
&= \frac{\sum_{i \text{ is selected}} F_i(1/d_i)}{\mathrm{UB}[N,K]} \cdot \mathrm{UB}[N,K] \cdot T - \mathcal{O}(N) \\
&\ge \gamma_K \cdot \mathrm{UB}[N,K] \cdot T - \mathcal{O}(N).
\end{aligned}$$

□

Proof of Lemma 5.

Let $\{x_i^*\}$ be an optimal solution of (2). From Lemma 2, we can assume that at most 1 of its non-zero components $x_{i_0}^*$ is not in $\{1/d_i^{(k)}\}_{k\ge 1}$. We round $x_{i_0}^*$ to $\widetilde{x}_{i_0}^* = \min\left\{y \ge x_{i_0}^* : 1/y \in \{d_i^{(k)}\}_{k\ge 1}\right\}$. We apply Step 1 in Section 3.2 to $\{x_i^*\}_{i \ne i_0} \cup \{\widetilde{x}_{i_0}^*\}$ and obtain $\{1/d_i\}$ such that $d_i \in \mathcal{D}[a]$. Define $\{x_{i,j,d}\}$ as follows,

$$x_{i,j,d} = \mathbb{1}\{d = d_i\}.$$

Then $\{x_{i,j,d}\}$ satisfies the constraints of (4), which means it is a feasible solution. Thus, the optimal objective value of (4) is

lower bounded by

$$\begin{aligned}
&\sum_{i\in[N]}\sum_{d\in\mathcal{D}_\phi[a]}\hat{R}_{i,j}(d)x_{i,j,d}/d\\
\geq&\sum_{i\in[N],d_i\in\mathcal{D}_\phi[a]}R_i(d_i)/d_i\\
\geq&\sum_{i\in[N]}R_i(d_i)/d_i-\sum_{i\in[N],d_i\notin\mathcal{D}_\phi[a]}R_i(d_i)/d_i\\
\geq&\sum_{i\in[N],i\neq i_0}\frac{a}{a+1}F_i(x_i^*)+\frac{a}{a+1}F_i(\widetilde{x}_{i_0}^*)-N\frac{R_{\max}}{\phi/2}\\
\geq&\sum_{i\in[N]}\frac{a}{a+1}F_i(x_i^*)-\frac{2NR_{\max}}{\phi}\\
=&\frac{a}{a+1}\mathrm{UB}[N,K]-\frac{2NR_{\max}}{\phi},
\end{aligned}$$

where the third inequality follows from Lemma 3.

□

Proof of Lemma 6. Consider the following more general problem.

$$\begin{aligned}
\max_x\quad&\sum_{i\in[N]}\sum_{s\in\mathcal{S}}r_{i,s}x_{i,s}\\
\text{s.t.}\quad&\sum_{i\in[N]}\sum_{s\in\mathcal{S}}w_{i,s}x_{i,s}\leq K',\\
&\sum_{s\in\mathcal{S}}x_{i,s}\leq 1,\quad\forall i\in[N],\\
&x_{i,s}\in\{0,1\},\quad\forall i\in[N],\forall s\in\mathcal{S}.
\end{aligned}\tag{2}$$

We first assume that $r_{i,s}\in\mathbb{Z}_+\cup\{0\}$, then we let $v(n,r)$ be the value of following problem.

$$\begin{aligned}
\min_x\quad&\sum_{i\in[n]}\sum_{s\in\mathcal{S}}w_{i,s}x_{i,s}\\
\text{s.t.}\quad&\sum_{i\in[n]}\sum_{s\in\mathcal{S}}r_{i,s}x_{i,s}=r,\\
&\sum_{s\in\mathcal{S}}x_{i,s}\leq 1,\quad\forall i\in[n],\\
&x_{i,s}\in\{0,1\},\quad\forall i\in[n],\forall s\in\mathcal{S}.
\end{aligned}$$

If the problem is infeasible, we let $v(n,r)=+\infty$, then we have the following recurrence formula:

$$v(n,r)=\min_{s:r_{n,s}\leq r}\left\{v(n-1,r),w_{n,s}+v(n-1,r-r_{n,s})\right\}.$$

We also have the initial conditions:

$$v(1,r)=\begin{cases}w_{1,s}, & \text{if } r=r_{1,s},\\ +\infty, & \text{else},\end{cases}\qquad v(n,0)=0,\quad\forall n\in[N].$$

Let $r_{\max}=\max_{i,s}r_{i,s}$, then the largest possible r is $Nr_{\max}$. Thus, $v(n,r)$ can be computed within $\mathcal{O}\left(N^2r_{\max}|\mathcal{S}|\right)$ time. The maximal reward for (2) is then computed by iterating through $\{v(n,r)\}_{n\in[N],r\leq Nr_{\max}}$ such that $v(n,r)\leq K'$ while r is maximized. Finding the optimal solution requires tracing back $v(n,r)$ to the initial conditions, which consumes $\mathcal{O}(N|\mathcal{S}|)$ time. Thus, the total time complexity is $\mathcal{O}\left(N^2r_{\max}|\mathcal{S}|\right)$.

For the general case, we define $\tilde{r}_{i,s} = \lfloor \frac{Nr_{i,s}}{\epsilon r_{\max}} \rfloor$. We compute (2) with $\{r_{i,s}\}$ replaced by $\{\tilde{r}_{i,s}\}$. Then since

$$0 \leq \frac{\frac{N}{\epsilon r_{\max}} r_{i,s} - \tilde{r}_{i,s}}{\frac{N}{\epsilon r_{\max}} r_{i,s}} \leq \frac{1}{\frac{Nr_{i,s}}{\epsilon r_{\max}}} \leq \epsilon, \quad \forall i \in [N], s \in \mathcal{S},$$

The solution we compute is a $(1-\epsilon)$-optimal solution of (2). As a final step, we substitute $\mathcal{S}$ with $\mathcal{D}_\phi[a]$, and the computation time is

$$\mathcal{O}\left(N^2 \max_{i,s} \left\lfloor \frac{Nr_{i,s}}{\epsilon r_{\max}} \right\rfloor |\mathcal{D}_\phi[a]|\right) = \mathcal{O}\left(\frac{N^3 a \log_2 \phi}{\epsilon}\right).$$

□

Proof of Theorem 2.

For completeness, we restate some definitions. Let $n_{i,j}(d)$ be the number of samples we have collected for $R_i(d)$ from the beginning of the whole time horizon to the end of phase j. Here we let $n_{i,0}(d) = 0$ for all $i \in [N]$ and $d \in \mathbb{Z}_+$. At the beginning, we have a natural upper bound $R_i(d) \leq R_{\max}$. Let

$$\bar{R}_{i,j-1}(d) \triangleq \frac{\sum_{\ell=1}^{n_{i,j-1}(d)} \hat{R}_i^\ell(d)}{n_{i,j-1}(d)}$$

be the empirical mean of $R_i(d)$ calculated by the samples collected prior to phase j. Here, $\hat{R}_i^\ell(d)$ is the ℓth sampled reward we collected when we offer product i d time periods after we offered it last time. The upper bound $\hat{R}_{i,j}(d)$ is then computed by

$$\min\left\{\bar{R}_{i,j-1}(d) + R_{\max}\sqrt{\frac{2\log(KT)}{\max\{n_{i,j-1}(d),1\}}},\ R_{\max}\right\}.$$

Let $\mathcal{G}$ be the "good event" that $\forall i \in [N]$, all phases j and all $d \in \mathcal{D}_\phi[a]$, the following holds:

$$\left|R_i(d) - \bar{R}_{i,j-1}(d)\right| \leq R_{\max}\sqrt{\frac{2\log(KT)}{\max\{n_{i,j-1}(d),1\}}}. \tag{3}$$

Since $\forall i \in [N]$, after each phase, we update the estimation of at most one element in $\{R_i(d)\}_{d\in\mathcal{D}_\phi[a]}$, and so by Hoeffding's inequality,

$$\begin{aligned}\mathbb{P}\left(\mathcal{G}^c\right) &\leq N \cdot \left\lceil \frac{T}{\phi} \right\rceil \cdot 2\exp\left(-2 \cdot 2\log(KT)\right) \\ &\leq N \cdot \frac{2T}{\phi} \cdot 2\exp\left(-4\log(KT)\right) \leq \frac{4N}{\phi K T^3}.\end{aligned}$$

Thus, the total loss incurred when $\mathcal{G}^c$ occurs is bounded by

$$\frac{4N}{\phi K T^3} R_{\max} KT = \mathcal{O}\left(\frac{NR_{\max}}{\phi T^2}\right).$$

Next, we consider the situation when $\mathcal{G}$ holds. Then we have

$$\hat{R}_{i,j}(d) \geq R_i(d) \geq \hat{R}_{i,j}(d) - 2R_{\max}\sqrt{\frac{2\log(KT)}{\max\{n_{i,j-1}(d),1\}}}$$

because of (3) and $R_i(d) \le R_{\max}$. For brevity, we write $\gamma_{K,\epsilon} = \gamma_K(1-\epsilon)$. The reward obtained at a given phase j is

$$\begin{aligned}
&\phi \sum_{i=1}^{N} R_i(d_{i,j})/d_{i,j} - \mathcal{O}(NR_{\max}) \\
&= \phi\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \phi\left(\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \sum_{i=1}^{N} R_i(d_{i,j})/d_{i,j}\right) - \mathcal{O}(NR_{\max}) \\
&\ge \phi\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \phi\sum_{i=1}^{N}\left(\hat{R}_{i,j}(d_{i,j})/d_{i,j} - R_i(d_{i,j})/d_{i,j}\right) - 2NR_{\max} - \mathcal{O}(NR_{\max}) \\
&\ge \phi\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \phi\sum_{i=1}^{N}\mathcal{O}\left(\frac{R_{\max}\sqrt{\log(KT)}}{d_{i,j}\sqrt{\max\{n_{i,j-1}(d_{i,j}),1\}}}\right) - \mathcal{O}(NR_{\max}),
\end{aligned}$$

where the first inequality is from (5). Summing over all phases, we can derive that the reward obtained under $\mathcal{G}$ is lower bounded by

$$\begin{aligned}
&T\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \sum_{i=1}^{N}\sum_{j}\mathcal{O}\left(\frac{\phi R_{\max}\sqrt{\log(KT)}}{d_{i,j}\sqrt{\max\{n_{i,j-1}(d_{i,j}),1\}}}\right) - \mathcal{O}\left(\frac{NR_{\max}T}{\phi}\right) \\
&= T\gamma_{K,\epsilon}\mathrm{UB}[N,K] - \sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]}\sum_{j:d_{i,j}=d}\mathcal{O}\left(\frac{\phi R_{\max}\sqrt{\log(KT)}}{d_{i,j}\sqrt{\max\{n_{i,j-1}(d_{i,j}),1\}}}\right) - \mathcal{O}\left(\frac{NR_{\max}T}{\phi}\right)
\end{aligned}$$

Note that for all j such that $d_{i,j} = d$, we can list in the increasing order $\{j_0, j_1, j_2, \cdots\}$ ($j_0 = 0$), and we have

$$n_{i,j_\ell-1}(d) \ge n_{i,j_{\ell-1}}(d) \ge n_{i,j_{\ell-1}-1}(d) + \left\lceil\frac{\phi}{d}\right\rceil - 1 \ge n_{i,j_{\ell-1}-1}(d) + \frac{\phi}{3d} \ge \frac{\phi}{3d}, \quad \forall \ell \ge 2.$$

Thus, we have

$$\frac{\phi}{d\sqrt{\max\{n_{i,j_2-1}(d),1\}}} \le \frac{\phi}{d\sqrt{n_{i,j_2-1}(d)}} \le 3\sqrt{n_{i,j_2-1}(d)},$$

and

$$\frac{\phi}{d\sqrt{\max\{n_{i,j_\ell-1}(d),1\}}} \le 3\frac{n_{i,j_\ell-1}(d) - n_{i,j_{\ell-1}-1}(d)}{\sqrt{n_{i,j_\ell-1}(d)}} \le 6\frac{n_{i,j_\ell-1}(d) - n_{i,j_{\ell-1}-1}(d)}{\sqrt{n_{i,j_\ell-1}(d)} + \sqrt{n_{i,j_{\ell-1}-1}(d)}}$$

Then we have

$$\begin{aligned}
&\sum_{j:d_{i,j}=d}\mathcal{O}\left(\frac{\phi}{d_{i,j}\sqrt{\max\{n_{i,j-1}(d_{i,j}),1\}}}\right) \\
&= \sum_{\ell}\mathcal{O}\left(\frac{\phi}{d\sqrt{\max\{n_{i,j_\ell-1}(d),1\}}}\right) \\
&= \sum_{\ell=1}\mathcal{O}\left(\frac{\phi}{d\sqrt{\max\{n_{i,j_\ell-1}(d),1\}}}\right) + \sum_{\ell=2}\mathcal{O}\left(\frac{\phi}{d\sqrt{\max\{n_{i,j_\ell-1}(d),1\}}}\right) + \sum_{\ell\ge3}\mathcal{O}\left(\frac{\phi}{d\sqrt{\max\{n_{i,j_\ell-1}(d),1\}}}\right) \\
&\le \mathcal{O}\left(\frac{\phi}{d}\right) + \mathcal{O}\left(\sqrt{n_{i,j_2-1}(d)}\right) + \sum_{\ell\ge3}\mathcal{O}\left(\frac{n_{i,j_\ell-1}(d) - n_{i,j_{\ell-1}-1}(d)}{\sqrt{n_{i,j_\ell-1}(d)} + \sqrt{n_{i,j_{\ell-1}-1}(d)}}\right) \\
&\le \mathcal{O}\left(\frac{\phi}{d}\right) + \mathcal{O}\left(\sqrt{n_i(d)}\right),
\end{aligned}$$

where $n_i(d)$ is the number of samples we collected for $R_i(d)$ throughout the whole time horizon. Combined with the loss incurred under $\mathcal{G}^c$, the overall reward can be further bounded by

$$\begin{aligned}
&T\gamma_{K,\epsilon}\text{UB}[N,K] - \sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]}\sum_{j:d_{i,j}=d}\mathcal{O}\left(\frac{\phi R_{\max}\sqrt{\log(KT)}}{d_{i,j}\sqrt{\max\{n_{i,j-1}(d_{i,j}),1\}}}\right) - \mathcal{O}\left(\frac{NR_{\max}T}{\phi}\right)\\
\geq\ &T\gamma_{K,\epsilon}\text{UB}[N,K] - R_{\max}\sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]}\mathcal{O}\left(\frac{\phi\sqrt{\log(KT)}}{d} + \sqrt{n_i(d)\log(KT)}\right) - \mathcal{O}\left(\frac{NR_{\max}T}{\phi}\right)\\
\geq\ &T\gamma_{K,\epsilon}\text{UB}[N,K] - R_{\max}\mathcal{O}\left(\phi N\log(K+1)\sqrt{\log(KT)} + \sqrt{NTK^{\frac{3}{2}}\log\phi\log(KT)} + \frac{NT}{\phi}\right),
\end{aligned}$$

where in the last inequality we use

$$\sum_{d\in\mathcal{D}_\phi[a^*]}\frac{1}{d} \leq \sum_{a=1}^{a^*}\sum_{d\in\mathcal{D}_a}\frac{1}{d} \leq \sum_{a=1}^{a^*}\frac{2}{a} = \mathcal{O}\left(\log(K+1)\right),$$

and

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]}\sqrt{n_i(d)} &\leq \sqrt{\sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]} n_i(d).}\sqrt{\sum_{i=1}^{N}\sum_{d\in\mathcal{D}_\phi[a^*]}1}\\
&\leq \sqrt{KT\cdot N\left|\mathcal{D}_\phi[a^*]\right|} = \mathcal{O}\left(\sqrt{NTK^{\frac{3}{2}}\log\phi}\right).
\end{aligned}$$

With $\phi = \Theta\left(\sqrt{\frac{T}{\log(K+1)}}\right)$ and $\epsilon = \Theta\left(T^{-\frac{1}{2}}\right)$, the total reward is lower bounded by

$$T\gamma_K\text{UB}[N,K] - \mathcal{O}\left(\max\{N, N^{\frac{1}{2}}K^{\frac{3}{4}}\}R_{\max}\sqrt{T\log(K+1)\log T\log(KT)}\right),$$

since

$$T\epsilon\text{UB}[N,K] = \mathcal{O}\left(\epsilon T\cdot KR_{\max}\right) = \mathcal{O}\left(KR_{\max}\sqrt{T}\right) = \mathcal{O}\left(\max\{N, N^{\frac{1}{2}}K^{\frac{3}{4}}\}R_{\max}\sqrt{T}\right).$$

□

References

Bar-Noy, A., Bhatia, R., Naor, J., and Schieber, B. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27(3):518–544, 2002.

Cella, L. and Cesa-Bianchi, N. Stochastic bandits with delay-dependent payoffs. In *International Conference on Artificial Intelligence and Statistics*, pp. 1168–1177, 2020.