Online Contention Resolution Schemes for Network Revenue Management and Combinatorial Auctions

Will Ma[1], Calum MacRury[2], and Jingwei Zhang[3]

[1]Graduate School of Business and Data Science Institute, Columbia University, New York, NY 10027
wm2428@gsb.columbia.edu
[2]Graduate School of Business, Columbia University, New York, NY 10027
cm4379@columbia.edu
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
zhangjingwei@cuhk.edu.cn

Abstract

In the Network Revenue Management (NRM) problem, products composed of up to L resources are sold to stochastically arriving customers. We take a randomized rounding approach to NRM, motivated by developments in Online Contention Resolution Schemes (OCRS). The goal is to take a fractional solution to NRM that satisfies the resource constraints in expectation, and implement it in an online policy that satisfies the resource constraints in any state, while (approximately) preserving all of the sales that were prescribed by the fractional solution.

OCRS cannot be naively applied to NRM or revenue management problems in general, because customer substitution induces a negative correlation in products being demanded. We start by deriving an OCRS that achieves a guarantee of $1/(1+L)$ for NRM with customer substitution, matching a common benchmark in the literature. We then show how to beat this benchmark for all integers $L > 1$ assuming no substitution, i.e. in the standard OCRS setting. By contrast, we show that this benchmark is unbeatable using OCRS or any fractional relaxation if there is customer substitution, for all integers L that are the power of a prime number. Finally, we show how to beat $1/(1+L)$ even with customer substitution, if the products comprise one item from each of up to L groups.

Our results have corresponding implications for Online Combinatorial Auctions, in which buyers bid for bundles of up to L items, and buyers being single-minded is akin to no substitution. Our final result also beats $1/(1+L)$ for Prophet Inequality on the intersection of L partition matroids. All in all, our paper provides a unifying framework for applying OCRS to these problems, delineating the impact of substitution, and establishing a separation between the guarantees achievable with vs. without substitution under general resource constraints parametrized by L.

1 Introduction

In the Network Revenue Management (NRM) problem, a set of items M is sold in the form of *products*. Each product $j \in N$ has a fixed price r_j and represents a bundle of items $A_j \subseteq M$, with the constraint that the same item cannot be sold more than once. Over time $t = 1, \dots, T$, customers make independence stochastic choices about which product to purchase, which can be influenced by an online algorithm that dynamically controls product availability. All probability distributions governing customers and their choices are given in advance, and the objective of an online algorithm is to maximize its expected total revenue over the time horizon.

A common approach to NRM is to first solve a Linear Programming (LP) relaxation to obtain an optimal offline fractional solution, in which x_j prescribes the probability that each product j should be sold. The goal is then to "round" this solution, which is only feasible in expectation, to online decisions that respect the item feasibility constraints with probability (w.p.) 1. The easiest way to ensure the quality of the online decisions is to provide a *uniform* guarantee, where every product j is sold w.p. αx_j for some constant $\alpha \in [0, 1]$, guaranteeing an α-fraction of the optimal revenue. Online Contention Resolution Schemes (OCRS's) are designed to provide exactly this type of uniform guarantee, which have been shown to not worsen the best-possible α (Lee and Singla, 2018). OCRS's operate in an abstract setting: the products j are presented in sequence, with each one being "active" independently w.p. x_j, where "active" represents that a customer is willing to purchase product j. The OCRS must immediately decide whether to "accept" any active j, which implies selling the product. The OCRS may not want to accept all products that are active and feasible, because items should be preserved so that products j' appearing at the end of the sequence are still sold w.p. $\alpha x_{j'}$.

OCRS's cannot be directly applied to revenue management problems because the products j being active are not quite independent. Indeed, even though choices are assumed to be independent across customers, a particular customer t choosing one product j means that they would not choose another, inducing a form of negative correlation. Nonetheless, this is generally not worse than the typical independent setting of OCRS—for feasibility structures defined by matroids and knapsacks, algorithms have already been extended to handle this basic form of negative correlation, with identical guarantees α (see Subsection 1.3). Consequently, the subtlety with this basic form of negative correlation has been largely ignored.

In this paper, we show that this basic negative correlation can make the best-possible guarantee *strictly worse*, under general feasibility constraints parametrized by the maximum number of items in a product. This is a standard parametrization in NRM, and motivates us to define an extended notion of OCRS that handles this basic form of negative correlation, which we interpret as a random product being chosen at each time $t = 1, \dots, T$.

Definition 1.1 (Random-element OCRS). A universe of elements N is partitioned into disjoint subsets $N_1, \dots, N_T$, where each N_t is referred to as a batch. The OCRS is given a fractional solution $(x_j)_{j \in N}$, which satisfies both the feasibility constraints on N in expectation, and $\sum_{j \in N_t} x_j \leq 1$ for all t. Sequentially over $t = 1, \dots, T$, at most one random element from N_t is drawn to be active following probability vector $(x_j)_{j \in N_t}$, where no element is active w.p. $1 - \sum_{j \in N_t} x_j$. The OCRS must immediately decide whether to accept any active element, subject to the feasibility constraints. We say that the OCRS is *random-element α-selectable* if it guarantees to accept every element j w.p. αx_j, for all feasibility structures in some class, all choices of T and partitionings $N = N_1 \cup \cdots \cup N_T$, and all fractional solutions $(x_j)_{j \in N}$ satisfying both the feasibility constraints on N in expectation and $\sum_{j \in N_t} x_j \leq 1$ for all t.

Remark 1.1. We make the following remarks about Definition 1.1:

1. In the standard notion of OCRS, the guarantee only has to hold for the trivial partitioning where $T = |N|$ and $|N_1| = \cdots = |N_T| = 1$. There is typically no index t nor notion of time.
2. In random-element OCRS, we typically interpret t as time, and hence for simplicity we assume the batches arrive in order $N_1, N_2, \dots, N_T$. It can be checked that all of our positive results continue to hold if the t's arrive in an order chosen by an oblivious adversary, who knows the algorithm but not any random realizations.

Hereafter we focus on feasibility structures defined by *L-bounded products*. That is, there is a set of items M assumed to have one copy each (we explain why this assumption is without loss in Section 5). There is a set of products N, with each product j requiring a bundle of items $A_j \subseteq M$. An active product j is feasible to accept if and only if A_j does not intersect with $A_{j'}$ for any previously-accepted product j'. Recalling that x_j represents the probability of selling each product

$j \in N$, we say that $(x_j)_{j\in N}$ satisfies the feasibility constraints in expectation if

$$\sum_{j:i\in A_j} x_j \leq 1 \qquad\qquad \forall i \in M; \tag{1}$$

i.e. no item is required more than once in expectation. We want the OCRS guarantee α to hold for all feasibility structures where $1 \leq |A_j| \leq L$ for all $j \in N$, with no assumptions on the number of items/products or the bundles A_j otherwise. We note that it is possible for different products to require the same set of items, i.e. $A_j = A_{j'}$ for $j \neq j'$. We allow the guarantee α to depend on L, which is treated as a constant.

If $L = 1$, then we are in a classical (non-network) revenue management setting where only one item can be sold at a time. If $L = 2$, then items and products can be interpreted as vertices and edges in a graph respectively, where a set of products is feasible to sell if and only if they form a matching in the graph. This is a well-studied setting, with (1) being the matching polytope. (Technically our formulation is more general by allowing for single-vertex products and parallel edges; nonetheless, it can be checked that the positive results from this literature continue to hold for these cases.) In this setting, Ezra et al. (2022) have considered random-element OCRS where the guarantee only has to hold under a specific partitioning (vertex-arrival "batches"), and shown how the guarantee can improve. By contrast, we study how the guarantee can worsen under a worst-case partitioning.

1.1 Results for (Random-element) OCRS with L-bounded Products

A simple random-element $1/(1+L)$-selectable OCRS (Section 2). We warm up by deriving a simple random-element $1/(1+L)$-selectable OCRS, based on the idea of exact selection from Ezra et al. (2022). This implies a guarantee of $1/(1+L)$ relative to the LP relaxation (and optimal dynamic program) in NRM problems with general pricing and assortment controls, as long as each product contains at most L items. This also implies a guarantee of $1/(1+L)$ relative to the prophet's welfare in general Online Combinatorial Auctions (OCA), as long as each agent wants most L items. We defer the full descriptions of these problems, and their reductions to random-element OCRS, to Section 5.

We note that the guarantee of $1/(1+L)$ was already known in both the NRM (Ma et al., 2020) and OCA (Correa et al., 2023) problems, with the latter result being achieved by a particularly simple static item pricing mechanism. Both of the $1/(1+L)$ results in Ma et al. (2020) and Correa et al. (2023) have been extended in subsequent works, as we discuss in Subsection 1.3. Therefore, $1/(1+L)$ can be viewed as a benchmark to beat for L-bounded products.

Beating $1/(1+L)$ in standard OCRS (Subsection 4.1). We establish a guarantee strictly exceeding $1/(1+L)$ for all $L > 1$ in the standard OCRS setting without our notion of random elements. We note that a guarantee strictly exceeding 1/3 was already known in the $L = 2$ case which corresponds to matchings in graphs (Ezra et al., 2022; MacRury et al., 2023), but their "witness" arguments do not easily extend to a general $L > 2$. Indeed, as we explain in Subsection 1.2, we use a new analysis technique that also sheds new light even for the $L = 2$ case.

Standard OCRS can still be applied to the NRM problem with independent time-varying Poisson demands, which is the original case of NRM considered in Gallego and Van Ryzin (1997). They can also be applied to the OCA problem with single-minded agents, a case of interest in Correa et al. (2023); Marinkovic et al. (2023). We explain these special cases in Section 5, and our result implies a guarantee strictly exceeding $1/(1+L)$ for both of them.

Unbeatability of $1/(1+L)$ in random-element OCRS (Section 3). We show that the guarantee of $1/(1+L)$ is best-possible for general L under our notion of random-element OCRS. In particular, we show how to translate a finite affine plane of order L into an instance with random L-bounded elements in which no OCRS can be better than $1/(1+L)$-selectable. Finite affine planes are known to exist when L is a prime power, i.e. $L = p^k$ for some prime number p and positive integer k (see Moorhouse (2007) for a reference). They are known to not exist for $L = 6, 10$, but otherwise the problem is open. In sum, our result implies that $1/(1+L)$ is unbeatable for random-element OCRS when $L = 2, 3, 4, 5, 7, 8, 9, 11$, and possibly 12.

The main idea behind our construction is to partition the elements into "intersecting" perfect matchings, and we elaborate on the connection with finite affine planes in Section 3. The $L = 2$ result implies that OCRS's for graph matching cannot be better than 1/3-selectable when edges are batched adversarially, something not previously known. Our result more generally shows that $1/(1+L)$ is unbeatable by any analysis that does not discriminate between the LP relaxation (also known as the fluid or ex-ante relaxation) vs. tighter ones, which is the case for all NRM analyses known to us. We remark that for OCA, Correa et al. (2023) show that $1/(1+L)$ is unbeatable against the tighter prophet benchmark, if the algorithm is restricted to using static item prices. Their result also does not require any primality assumptions on L.

Beating $1/(1+L)$ in random-element OCRS (Subsection 4.2). Finally, we show that $1/(1+L)$ can be beaten even for random-element OCRS on L-partite hypergraphs, where the items come from L groups and each product requires at most one item from each group. This is a natural setting in NRM where each product can e.g. be a "combo" of a main dish + side + drink, and also applies to hotel bookings for intervals of length at most L (Rusmevichientong et al., 2023). Moreover, this captures the prophet inequality problem on the intersection of L partition matroids, whose tight ratio is mentioned as an open problem in Correa et al. (2023). They show that the tight ratio is at least $1/(1+L)$; we now show it is strictly bigger. Moreover, our work generally suggests that the tight ratio could depend on whether the elements in the prophet inequality problem are allowed to be random.

1.2 Techniques

Warm-up: attaining $1/(1+L)$. Our random-element OCRS is based on the idea of exact selection, first used by Ezra et al. (2022) for standard OCRS on graphs. To get our $\alpha = 1/(L+1)$-selectable random-element OCRS, we extend the idea of exact selection to arbitrary batches and values of L. The idea is to describe the random-element OCRS recursively in terms of the T batches: Assuming each product $j' \in N_{t'}$ is selected w.p. $\alpha x_{j'}$ for all $t' < t$, we extend this guarantee to batch N_t. This requires selecting an active $j \in N_t$ w.p. $\alpha/\mathbb{P}(\text{every item of } j \text{ is available})$, as so the crux of the analysis is arguing that this is well-defined, i.e., $\alpha \leq \mathbb{P}(\text{every item of } j \text{ is available})$. Our $1/(L+1)$ guarantee applies a simple union bound over the L items of product j, which combined with the feasibility constraint (1) yields the desired inequality.

Beating $1/(1+L)$. In order to beat $1/(1+L)$, the problem boils down to improving on the union bound, which can underestimate the probability that an incoming product j is feasible. In existing works studying the standard OCRS (e.g., Ezra et al. (2022) and MacRury et al. (2023)) for $L=2$, this is done via a *witness argument.* In this setting, a product j contains two items, and the goal is to lower bound the *joint* probability that both items are selected before j arrives. Since characterizing this exact probability for an arbitrary input is intractable, previous works have instead focused on defining a *witness* event that implies both items are selected, yet whose probability can be estimated. Unfortunately, these witness events heavily rely on the graph structure of $L = 2$, and the fact that all batches contain a single product. They do not seem easy to generalize to $L \geq 3$, nor to when there are correlations between products due to the batches.

We develop a new framework aimed at enhancing the guarantee. To improve the union bound, it is sufficient to demonstrate the existence of a strictly positive probability of the intersection of certain events. In our setting, where we consider a set of items, our goal is to analyze the cumulative probabilities of any two items being unavailable by the end across all possible combinations. This probability can be further lower bounded by the summation of probabilities that pairs of products are accepted across all possible pairs. The framework consists of two key steps. First, for any two products j, j' belonging to distinct batches $N_t, N_{t'}$ that also have disjoint item sets $A_j \cap A_{j'} = \emptyset$, we show (in Lemma 4.1 in Section 4) that

$$\mathbb{P}\left(j \text{ accepted} \cap j' \text{ accepted}\right) \geq C x_j x_{j'} \tag{2}$$

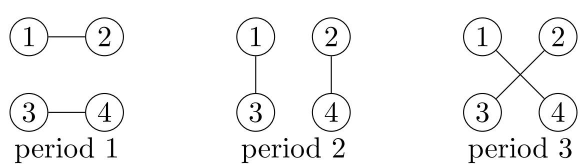
where $C > 0$ is a constant. To the best of our knowledge, this fact was not apparent from Ezra et al. (2022); MacRury et al. (2023): it says that for *every pair of disjoint edges*, the OCRS of Ezra et al. (2022) has positive probability of accepting both of them. We prove (2) by reducing it to a concave optimization problem, in which the coefficient matrix for constraints is totally unimodular and thus the optimal solution can be explicitly characterized.

Second, we leverage (2) to show that multiple bad events for a newly arriving product can occur, and hence the union bound is not tight. Indeed, suppose product j_0 is newly arriving and $A_{j_0} = \{1, \ldots, L\}$. In this case, (2) says that if $A_j, A_{j'}$ both intersect A_{j_0} but are themselves disjoint, and moreover come from different batches and have $x_j, x_{j'} > 0$, then both of the bad events of j being accepted and j' being accepted (either of which would make j_0 infeasible) can occur. Eventually this reduces to an adversary's problem of minimizing

$$\sum_{i=1}^{L}\sum_{i'\neq i}\sum_{t=1}^{T}\sum_{t'\neq t}\sum_{\substack{j\in N_t, j'\in N_{t'}:\\ A_j\cap A_{j'}=\emptyset\\ i\in A_j, i'\in A_{j'}}} x_j x_{j'} \tag{3}$$

subject to $\sum_{j:i\in A_j} x_j = 1$ for all $i = 1,\ldots,L$ and $|A_j \cap \{1,\ldots,L\}| \leq 1$ for all j. If the adversary can construct an arbitrary item-product configuration and with arbitrary batches, then they can indeed achieve an objective value of 0 in (3) (which corresponds to the construction in our negative result), and multiple bad events cannot occur. However, if we restrict the adversary to the standard OCRS setting (i.e. $|N_t| = 1$ for all t), or restrict the item-product configuration to be an L-partite hypergraph, then (3) is lower-bounded by a non-zero constant (see Lemma 4.2 in Subsection 4.1, and Lemma 4.4 in Subsection 4.2). To bound the adversary's optimization problem we use the fact that every product intersects with $\{1,\ldots,L\}$ at most once and reduce (3) into a more compact form with a bilinear objective and linear constraints. Interestingly, we can characterize the optimal solution in the standard OCRS setting. Ultimately, this allows us to beat $1/(1+L)$ in either of these settings.

Unbeatability of $1/(1+L)$. To provide more intuition for the unbeatability, we provide an explicit counterexample for $L = 2$ here and show $1/3$ cannot be surpassed under random-element OCRS. In this illustrative example, there are 3 periods and 4 items: $\{1,2,3,4\}$. The figure below represents the possible products in each period, where each edge denotes one product:



For example, in the first period, there are two possible products: $(1,2)$ and $(3,4)$. If products are labeled by the items contained (i.e. the two endpoints of the edge), then this construction amounts to $N_1 = \{(1,2),(3,4)\}, N_2 = \{(1,3),(2,4)\}$, and $N_3 = \{(1,4),(2,3)\}$.

Additionally, the active probability of a product within the first two periods is $(1-\varepsilon)/2$ and the active probability of a product in the final period is ε, ensuring the feasibility constraint is satisfied. Formally, we have $x_{(1,2)} = x_{(3,4)} = x_{(1,3)} = x_{(2,4)} = (1-\varepsilon)/2$ and $x_{(1,4)} = x_{(2,3)} = \varepsilon$. We now explain why 1/3 is unbeatable in this example. Note that for the product $(1,4)$, the probability that this product is feasible is calculated as follows:

$$
\begin{aligned}
&\mathbb{P}\left(\text{both items 1 and 4 are available}\right) \\
=&1-\mathbb{P}(1 \text{ is used})-\mathbb{P}(4 \text{ is used})+\mathbb{P}\left(\text{both 1 and 4 are used}\right) \\
=&1-\alpha\left(x_{(1,2)}+x_{(1,3)}+x_{(3,4)}+x_{(2,4)}\right)+\mathbb{P}\left(\text{both 1 and 4 are used}\right) \\
=&1-2\alpha(1-\varepsilon)+\mathbb{P}\left(\text{both 1 and 4 are used}\right),
\end{aligned}
$$

where the second equality holds because the probability that a product j is accepted is αx_j under OCRS. Moreover, $\mathbb{P}\left(\text{both 1 and 4 are used}\right)=0$ because it is not possible for two distinct edges to be selected before period 3—the non-conflicting edges are in the same batch and hence cannot both be active. Therefore, for the OCRS to remain valid, it must hold that $1-2\alpha(1-\varepsilon)\geq\alpha$ for any $\varepsilon>0$, which implies $\alpha\leq 1/3$. Expanding this intuition to general L, we find that as long as there exists a finite affine plane of order L, we can make a similarly adversarial construction where the union bound is tight and $1/(1+L)$ is unbeatable. The construction here with $L=2$ is a special case of a finite affine plane of order 2, with 3 parallel classes of 2 lines each. We defer to Section 3 for further details.

1.3 Further Related Work

Random-element OCRS. Our notion of random elements, which imply a basic form of negative correlation, is not new to the vast literature on online Bayesian selection and allocation. That being said, we are the first to show that random elements can *worsen* the best-possible guarantee, which is why we explicitly distinguish between random-element OCRS and the standard OCRS with fixed elements.

Under the simplest online selection constraint where at most k elements can be accepted, there is no difference between fixed vs. random elements, because elements are identical. Under general matroid constraints (in which elements are non-identical), the prophet inequality of Kleinberg and Weinberg (2012) has been extended to handle random elements in the context of combinatorial auctions, with the same guarantee of 1/2 (Dutting et al., 2020). Similarly, the ex-ante matroid prophet inequality of Lee and Singla (2018) has been extended to handle random elements in the context of assortment optimization, also with the same guarantee of 1/2 (Baek and Ma, 2022). Under knapsack constraints, Jiang et al. (2022) establish a tight guarantee of $1/(3+e^{-2})$ for the OCRS problem, which they later extend to elements with random sizes, corresponding to random elements. In sum, for matroids and knapsacks, guarantees for fixed elements appear to extend to random elements, even though there is no black-box reduction.

Random elements can also be interpreted as a basic form of negative correlation. In this vein, classical prophet inequalities have been shown to extend to negatively dependent random variables (Rinott and Samuel-Cahn, 1987; Samuel-Cahn, 1991). Meanwhile, Dughmi (2020) shows that $(1-1/e)$-selectable offline contention resolution schemes for matroids can be extended under various forms of negative correlation, and even some cases of positive correlation.

Overall, the findings from the literature suggest that random elements and negative correlation should not worsen guarantees in online Bayesian selection. In stark contrast, our work finds that they do worsen guarantees for matchings in graphs, and more generally, L-bounded products.

Extensions of $1/(1+L)$ results. In NRM, the $1/(1+L)$ guarantee of Ma et al. (2020) has been extended to both reusable items (Baek and Ma, 2022) and flexible products (Zhu and Topaloglu, 2023). In OCA, the $1/(1+L)$ guarantee of Correa et al. (2023) has been shown to also hold when only a single sample is given about each distribution, if the arrival order is random (Marinkovic et al., 2023).

2 A Simple Random-element OCRS that is $1/(1+L)$-selectable

In this section, we design a simple random-element OCRS π which is α-selectable for $\alpha = 1/(L+1)$. For each $j \in N$, define X_j to be an indicator random variable for the event that j is active. We wish to design π in such a way that for each *time step* or *period* $t \in [T] = \{1, \dots, T\}$,

$$\mathbb{P}(j \text{ is accepted by } \pi \mid X_j = 1) = \alpha, \forall j \in N_t. \tag{4}$$

Let us first introduce some addition terminology and notation for an arbitrary OCRS π. Specifically, we say that a product $j \in N_t$ is *available* (i.e., free), provided $A_{j'} \cap A_j = \emptyset$ for each $j' \in N$ accepted by π before step t. Similarly, we say that an item $i \in M$ is *available at step* $1 \leq t \leq T$, provided

$i \notin A_{j'}$ for each $j' \in N$ accepted by π before step t. The choice of π will always be clear, so we denote these events by F_j and $F_{t,i}$, respectively. We also denote Z_j as the event that π accepts product j. Observe that F_j occurs if and only if $\cap_{i \in A_j} F_{t,i}$ occurs.

We now define π recursively in terms of $t \in [T]$. Specifically, for $t = 1$, π accepts an active product of N_1 (if any) independently w.p. α. For $t > 1$, assume that π is defined up until step $t-1$. We extend the definition of π to step t in the following way:

Definition 2.1. If $j \in N_t$ is active and available, then π accepts j independently w.p. $\min\{1, \alpha/\mathbb{P}(F_j)\}$.

Remark 2.1. Since the event F_j depends on the decisions of π strictly before step t, π is well-defined. Computing the exact value of $\mathbb{P}(F_j)$ is computationally challenging, however it can be estimated via Monte Carlo simulation. In Appendix A, we discuss the complexity of implementing the OCRS and provide the number of samples needed in order to achieve a given error tolerance.

Theorem 2.1. *If $\alpha = 1/(1+L)$, then π is an α-selectable random-element OCRS.*

Proof of Theorem 2.1. It suffices to verify (4) inductively. The base case of $t = 1$ clearly holds, so take $t > 1$, and assume that (4) holds for each $t' < t$. We verify (4) holds for t.

Fix an arbitrary $j \in N_t$. Observe that due to Definition 2.1, conditional on $X_j = 1$, j is accepted w.p. $\mathbb{P}(F_j) \cdot \min\{1, \frac{\alpha}{\mathbb{P}(F_j)}\}$. Thus, in order to complete the inductive step, we must argue that $\alpha \leq \mathbb{P}(F_j)$. Since $\alpha = 1/(L+1)$, it suffices to show that $\mathbb{P}(F_j) \geq 1 - \alpha L$. Now,

$$\mathbb{P}(F_j) = \mathbb{P}\left(\cap_{i \in A_j} F_{t,i}\right) = 1 - \mathbb{P}\left(\cup_{i \in A_j} \bar{F}_{t,i}\right) \geq 1 - \sum_{i \in A_j} \mathbb{P}\left(\bar{F}_{t,i}\right),$$

where $\bar{F}_{t,i}$ is the complement of $F_{t,i}$, and the final inequality uses a union bound. But, $\bar{F}_{t,i}$ occurs if and only if there exists some $t < t'$, $j' \in N_{t'}$ with $i \in A_{j'}$ for which $Z_{j'}$ occurs. Yet by (4),

$$\mathbb{P}(\bar{F}_{t,i}) = \sum_{t'<t} \sum_{j' \in N_{t'}: i \in A_{j'}} \mathbb{P}(Z_{j'}) = \alpha \sum_{t'<t} \sum_{j' \in N_{t'}: i \in A_{j'}} x_j' \leq \alpha,$$

where the inequality follows from the feasibility constraint (1). Thus,

$$\mathbb{P}(F_j) \geq 1 - \sum_{i \in A_j} \mathbb{P}\left(\bar{F}_{t,i}\right) \geq 1 - \alpha \sum_{i \in A_j} \sum_{j': i \in A_{j'}} x_{j'} \geq 1 - \alpha |A_j| \geq 1 - L\alpha,$$

and so the proof is complete. □

3 Unbeatability of $1/(1+L)$

As discussed in Section 1.2 for $L = 2$, no OCRS is better than 1/3-selectable. In this section, we generalize this hardness result to other values of L. In fact, we prove a stronger result that no online algorithm can attain a competitive ratio better than $1/(1+L)$ against the optimal value of a certain fluid LP. The value of an optimal solution to this fluid LP upper bounds (i.e., relaxes) an accept-reject version of the Network Revenue Management problem, and is a special case of the problem mentioned in the introduction. Specifically, in each step at most one product is drawn from a distribution, at which point the online algorithm must irrevocably accept or reject the product, subject to not violating item constraints. We include the details of the problem below.

Definition 3.1 (Accept-Reject NRM Problem)**.** Let M be a collection of items, where initially there is a single copy of each item. Products $j \in N$ have fixed rewards $r_j \geq 0$, require a non-empty subset of items $A_j \subseteq M$, and are partitioned into disjoint batches $N_1, \dots, N_T$, where $T \in \mathbb{N}$. In step $t = 1, \dots, T$, a random product $j \in N_t$ is independently drawn w.p. λ_j, where no product is drawn w.p. $1 - \sum_{j \in N_t} \lambda_j$. The online algorithm must then immediately decide whether or not to accept j, where j can be accepted only if all its associated items $i \in A_j$ are currently available (i.e., each previously accepted product j' satisfies $A_{j'} \cap A_j = \emptyset$). The online algorithm's goal is to maximize the expected cumulative reward of the products accepted.

In the reduced NRM problem, we benchmark the performance of an online algorithm against the expected cumulative reward of the optimal *offline* allocation (i.e., assuming full knowledge of the products drawn in the T steps). In order to upper bound (i.e., relax) this benchmark, we consider the following fluid LP:

Definition 3.2 (Fluid LP)**.**

$$\begin{aligned}
\max \ & \sum_j r_j x_j \\
\text{s.t.} \ & \sum_{j: i \in A_j} x_j \leq 1 \qquad \forall i \in M, \\
& 0 \leq x_j \leq \lambda_j \qquad \forall j \in N.
\end{aligned} \tag{5}$$

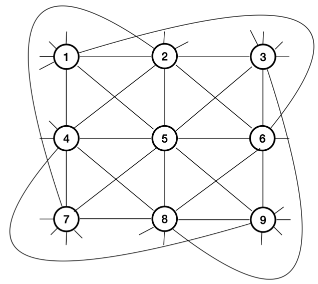


Figure 1: Finite affine plane with order 3.

To see that (5) is a relaxation, let x_j be the probability the benchmark accepts product j. Clearly, $x_j \leq \lambda_j$ for each $j \in N$, and $\sum_{j:i \in A_j} x_j \leq 1$ for each $i \in M$. Thus, $(x_j)_{j \in N}$ is a feasible solution to (5). Moreover, by using our random-element OCRS terminology and considering each product j to be active with probability x_j, an α-selectable random-element OCRS can be used to design an α-competitive online algorithm against the fluid LP. We defer the details of this argument, as we prove a much more general reduction in Theorem 5.1 of Section 5 which includes this argument as a special case. We are now ready to state our hardness result.

Theorem 3.1. *No online algorithm is better than* $1/(1+L)$*-competitive against* (5) *when* L *is a prime power.*

Corollary 3.2 (implied by Theorems 3.1 and 5.1)**.** *No random-element OCRS is better than* $1/(1+L)$*-selectable when* L *is a prime power.*

To prove Theorem 3.1, we will use the construction of a finite affine plane.

Definition 3.3 (Finite Affine Plane)**.** In a finite affine plane of order L, there are L^2 points and $L(L+1)$ distinct lines, each containing exactly L points. These lines can be grouped into $L+1$ classes of L parallel lines each, where the lines within a class are mutually disjoint and collectively contain all L^2 points. Finally, any two lines from two different classes intersect at exactly one point.

We display the finite affine plane of order 3 in Figure 1. Finite affine planes can be constructed from a finite field whenever L is the power of a prime number, and we refer to Moorhouse (2007) for further background. We now construct a configuration of items, products, and time steps for NRM, based on a finite affine plane, that is difficult for online algorithms.

Definition 3.4 (NRM Configuration)**.** Construct an item for each point in the affine plane, so that $|M| = L^2$. Construct a product j for each line, where A_j consists of the items corresponding to the L points in that line. Construct a batch N_t for each class of parallel lines, consisting of the products corresponding to the L lines in that class. In sum, we have $|N| = L(L+1)$, with N being a disjoint union of the batches N_t for $t = 1, \ldots, L+1$.

By the properties in Definition 3.3, this NRM configuration satisfies the following:

(i) For each $t \in [L+1]$ and $j, j' \in N_t$, if $j \neq j'$, then $A_j \cap A_{j'} = \emptyset$;

(ii) For each $1 \leq t < t' \leq L+1$ and $j \in N_t$, $j' \in N_{t'}$ it holds that $|A_j \cap A_{j'}| = 1$.

We now prove Theorem 3.1 using the NRM configuration from Definition 3.4, which exists by virtue of Definition 3.3 whenever L is a prime power.

Proof of Theorem 3.1. Assuming L is a prime power, the NRM configuration in Definition 3.4 exists. In this case, take $\varepsilon < 1/L$. We first set the remaining parameters necessary to describe an input to the accept-reject NRM problem. For each $t \in [L+1]$ and $j \in N_t$, if $t \leq L$ then set $\lambda_j = (1-\varepsilon)/L$ and $r_j = 1$, else set $\lambda_j = \varepsilon$ and $r_j = 1/(\varepsilon L)$.

We argue that no online algorithm can attain a competitive ratio better than $1/(L+1)$ against the fluid LP on this input. First observe it is always possible to accept all products in the fluid relaxation. That is, if we set $x_j = \lambda_j$ for each product j, then for each $i \in M$,

$$\sum_{j:i\in A_j} x_j = \sum_{j:i\in A_j} \lambda_j = \sum_{t\in[L+1]} \sum_{j\in N_t:i\in A_j} \lambda_j = L \cdot \frac{1-\varepsilon}{L} + \varepsilon = 1,$$

where the penultimate equality holds because by (i) in Definition 3.4. The optimal value of the fluid LP is thus equal to

$$\sum_j r_j x_j = \sum_j r_j \lambda_j = L^2 \cdot \frac{1-\varepsilon}{L} + L \cdot \varepsilon \cdot \frac{1}{\varepsilon L} = 1 + L(1-\varepsilon).$$

Now, because of condition (ii) of Definition 3.4, it is impossible to accept more than one product. This is because any two products of *distinct* batches share an item, and there is only one copy of each item. On the other hand, any online algorithm which accepts at most one product has an expected reward of at most 1. To see this, observe that if it accepts a product j in one of the first L

batches, then $r_j = 1$, so this holds. Otherwise, it waits until the final batch, leading to an expected reward of $L \cdot \varepsilon \cdot \frac{1}{\varepsilon L} = 1$. In either case, the claim holds. By taking $\varepsilon \to 0$, this implies that no online algorithm can attain a competitive ratio better than $1/(1 + L)$.

$\square$

4 Beating $1/(1 + L)$ under Different Conditions

Despite the $1/(1 + L)$ selection guarantee being tight for a random-element OCRS in general, it is possible to improve on this guarantee in certain scenarios. This was previously observed for standard OCRS with $L = 2$ by Ezra et al. (2022); MacRury et al. (2023). In this section, we develop a general framework to improve on $1/(L+1)$ for an arbitrary value of L. We then demonstrate our framework in two settings: standard OCRS, and random-element OCRS with L-partite hypergraphs.

Recall the recursively defined random-element OCRS π of Section 2 which was parameterized by $\alpha \in [0, 1]$. Our general framework proceeds by considering the same OCRS, yet with $\alpha > 1/(L+1)$. The exact value of α will be set depending on whether we are working in the standard OCRS setting, or the random-element L-partite setting. In order to simplify the indices later, let us assume that there are $T + 1$ batches. For each $1 \leq t \leq T + 1$, we again define the induction hypothesis,

$$\mathbb{P}(Z_j \mid X_j = 1) = \alpha, \forall j \in N_t, \tag{6}$$

where Z_j is the event j is accepted by π. Observe that when verifying (6), we can assume without loss of generality that we are working with a product j_0 from the final batch N_{T+1} for which $A_{j_0} = \{1, \ldots, L\}$. Recalling the definition of π, it suffices to argue that $\mathbb{P}(F_{j_0}) \geq \alpha$, where F_{j_0} is the event j_0 is available. Now, since $F_{T+1,i}$ is the event that item i is available at step $T + 1$,

$$\mathbb{P}(F_{j_0}) = \mathbb{P}\left(\cap_{i=1}^{L} F_{T+1,i}\right) = 1 - \mathbb{P}(\cup_{i=1}^{L} \bar{F}_{T+1,i}), \tag{7}$$

In Theorem 2.1, we lower bounded (7) by applying a simple union bound to $\mathbb{P}(\cup_{i=1}^{L} \bar{F}_{T+1,i})$. In order to improve on this, we first argue that with respect to minimizing (7), (equivalently, maximizing $\mathbb{P}(\cup_{i=1}^{L} \bar{F}_{T,i})$), the worst-case input for π occurs when the feasibility constraints on the items

$\{1, \ldots, L\}$ of j_0 are tight:

$$\sum_{j: i \in A_j} x_j = 1, \qquad\qquad \forall i \in \{1, \ldots, L\}. \tag{8}$$

To justify this assumption, observe that if (8) does *not* hold for items $M' \subseteq \{1, \ldots, L\}$, then we can always consider an auxiliary input identical to the original one except with an additional product for each item of M', all of which arrive before time $T+1$. Due to the definition of π, it is clear that adding these products can only *increase* $\mathbb{P}(\cup_{i=1}^{L} \bar{F}_{T+1,i})$, and thus decrease (7). Finally, by a similar argument, the worst-case input for (7) occurs when $\sum_{j' \in N_{T+1}} x_{j'}$ is arbitrarily small. Thus, in the following computations we abuse notation slightly and write that $\sum_{j' \in N_{T+1}} x_{j'} = 0$, with the understanding that we actually mean $\sum_{j' \in N_{T+1}} x_{j'} \leq \varepsilon$ for some arbitrarily small constant $\varepsilon > 0$.

The remainder of our framework can be summarized in the following three steps:

(i) Using inclusion-exclusion, we lower bound (7) and improve on the union bound by considering an additional term that accounts for pairs of items not being available. This additional term can be further lower bounded by a sum over $\mathbb{P}(Z_j \cap Z_{j'})$ for certain products j, j'.

(ii) For any pair of products j, j' satisfying certain conditions, we show that $\mathbb{P}(Z_j \cap Z_{j'}) \geq C(\alpha) x_j x_{j'}$, where $C(\alpha, L)$ is some absolute constant, dependent only on α and L.

(iii) Combining steps (i) and (ii), the problem is reduced to lower bounding a sum over terms of the form $x_j x_{j'}$ (see (9)). This can then be reformulated as an optimization problem. For standard OCRS and random-element OCRS on L-partite hypergraphs, the optimal value of the optimization problem must be strictly positive, which allows us to beat $1/(L+1)$.

We being with step (i). We claim the following sequence of inequalities (with explanations following afterwards):

$$\begin{aligned}
\mathbb{P}\left(\cap_{i=1}^{L} F_{T+1,i}\right) \geq & 1 - \sum_{i=1}^{L} \mathbb{P}\left(\bar{F}_{T+1,i}\right) + \max_i \sum_{i' \neq i} \mathbb{P}\left(\bar{F}_{T+1,i} \cap \bar{F}_{T+1,i'}\right) \\
\geq & 1 - \sum_{i=1}^{L} \mathbb{P}\left(\bar{F}_{T+1,i}\right) + \frac{1}{L} \sum_{i=1}^{L} \sum_{i' \neq i} \mathbb{P}\left(\bar{F}_{T+1,i} \cap \bar{F}_{T+1,i'}\right) \\
\geq & 1 - \alpha L + \frac{1}{L} \sum_{i=1}^{L} \sum_{i' \neq i} \mathbb{P}\left(\bar{F}_{T+1,i} \cap \bar{F}_{T+1,i'}\right)
\end{aligned}$$

$$\geq 1-\alpha L+\frac{1}{L}\sum_{i=1}^{L}\sum_{i'\neq i}\left(\sum_{j:\{i,i'\}\subseteq A_j}\mathbb{P}(Z_j)+\sum_{\substack{t,t'\in[T]:\\ t\neq t'}}\sum_{\substack{j\in N_t,j'\in N_{t'}:\\ A_j\cap A_{j'}=\emptyset\\ A_j\cap[L]=i,A_{j'}\cap[L]=i'}}\mathbb{P}\left(Z_j\cap Z_{j'}\right)\right)$$

$$=1-\alpha L+\frac{1}{L}\sum_{i=1}^{L}\sum_{i'\neq i}\left(\sum_{j:\{i,i'\}\subseteq A_j}\alpha x_j+\sum_{\substack{t,t'\in[T]:\\ t\neq t'}}\sum_{\substack{j\in N_t,j'\in N_{t'}:\\ A_j\cap A_{j'}=\emptyset\\ A_j\cap[L]=i,A_{j'}\cap[L]=i'}}\mathbb{P}\left(Z_j\cap Z_{j'}\right)\right).$$

The first inequality follows by inclusion-exclusion, the second by an averaging argument, and the third by an application of the induction hypothesis (6) in the same way as done in the proof of Theorem 2.1. The fourth inequality holds by considering a subset of the events in which $\bar{F}_{T+1,i}\cap\bar{F}_{T+1,i'}$ holds, and the final inequality applies (6) again.

We now describe step (ii), where our goal is to lower bound $\mathbb{P}\left(Z_j\cap Z_{j'}\right)$ for $j\in N_t$ and $j'\in N_{t'}$, where $t\neq t'$. Recall that when π is presented a product j, it draws a random bit, say B_j, which is 1 independently w.p. $\min\{1,\alpha/\mathbb{P}(F_j)\}$ (note that indeed $\alpha/\mathbb{P}(F_j)\leq 1$, due to the induction hypothesis (6)). Let us say that j is *survives*, provided $B_jX_j=1$. Otherwise, we say that j is *dies*. Using this terminology, we describe a sufficient condition in order for $Z_j\cap Z_j'$ to occur. Specifically, suppose that each each product $j''\notin N_t\cup N_{t'}$ which shares an item with j or j'' dies. Then, $Z_j\cap Z_{j'}$ occurs, provided both j and j' survive. Using independence, the joint probability of these events is easily computed, and so we get that

$$\begin{aligned}
\mathbb{P}\left(Z_j\cap Z_{j'}\right)\geq&\frac{\alpha x_j}{\mathbb{P}\left(F_j\right)}\frac{\alpha x_{j'}}{\mathbb{P}\left(F_{j'}\right)}\prod_{\tau\notin\{t,t'\}}\left(1-\sum_{j''\in N_\tau:A_{j''}\cap\left(A_j\cup A_{j'}\right)\neq\emptyset}\frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})}\right)\\
\geq&\alpha^2x_jx_{j'}\prod_{\tau\notin\{t,t'\}}\left(1-\sum_{j''\in N_\tau:A_{j''}\cap\left(A_j\cup A_{j'}\right)\neq\emptyset}\frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})}\right)\\
\geq&\alpha^2x_jx_{j'}\prod_{\tau=1}^{T}\left(1-\sum_{j''\in N_\tau:A_{j''}\cap\left(A_j\cup A_{j'}\right)\neq\emptyset}\frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})}\right)
\end{aligned}$$

where the penultimate equality uses the trivial upper bound of 1 on $\mathbb{P}(F_j)$ and $\mathbb{P}(F_{j'})$, and the final inequality uses that each term in the product takes its value in $[0,1]$.

Lemma 4.1. *For any products j and j' with $A_j \cap A_{j'} = \emptyset$, it holds that*

$$\prod_{\tau=1}^{T} \left(1 - \sum_{j'' \in N_\tau : A_{j''} \cap \left(A_j \cup A_{j'} \right) \neq \emptyset} \frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})} \right) \geq \left(\frac{1 - \alpha(L+1) + \alpha/2L}{1 - \alpha L + \alpha/2L} \right)^{2L}.$$

The proof of Lemma 4.1 bounds each $\mathbb{P}(F_{j''})$ using the various $x_{j''}$ and then converts the product term into an expression depending only on the $x_{j''}$. By analyzing an optimization problem which minimizes the product term via the $x_{j''}$, we can then characterize the optimal solution, which leads to the result above. We provide a detailed proof in Appendix B.

By Lemma 4.1, in order to lower bound $\mathbb{P}(\cap_{i=1}^{L} F_{T+1,i})$, it remains to analyze

$$\sum_{i=1}^{L} \sum_{i' \neq i} \left(\sum_{j : \{i, i'\} \subseteq A_j} \alpha x_j + \alpha^2 \left(\frac{1 - \alpha(L+1) + \alpha/2L}{1 - \alpha L + \alpha/2L} \right)^{2L} \sum_{t=1}^{T} \sum_{t' \neq t} \sum_{\substack{j \in N_t, j' \in N_{t'} : \\ A_j \cap A_{j'} = \emptyset \\ A_j \cap [L] = i, A_{j'} \cap [L] = i'}} x_j x_{j'} \right).$$

We claim that in the worst case, $x_j = 0$ for any j such that $|A_j \cap [L]| \geq 2$. To see this, note that

$$\left(\frac{1 - \alpha(L+1) + \alpha/2L}{1 - \alpha L + \alpha/2L} \right)^{2L}$$

is decreasing in α. Thus, since $\alpha \geq 1/(1+L)$, this is upper bounded by $1/(2L+1)^{2L}$. Therefore, in order to minimize the summand for $\{i, i'\}$, it is never optimal to set $x_j > 0$ if $\{i, i'\} \subseteq A_j$. We can thus restrict our attention to the case where $|A_j \cap [L]| \leq 1$ for every product j. That is, we analyze

$$\sum_{i=1}^{L} \sum_{i' \neq i} \sum_{t=1}^{T} \sum_{t' \neq t} \sum_{\substack{j \in N_t, j' \in N_{t'} : \\ A_j \cap A_{j'} = \emptyset \\ i \in A_j, i' \in A_{j'}}} x_j x_{j'}, \ (\clubsuit) \tag{9}$$

subject to the constraints $\sum_{j : i \in A_j} x_j = 1$ for any $i \in [L]$ and $|A_j \cap [L]| \leq 1$ for any product j.

In general, ($\clubsuit$) can be as small as zero even with these two constraints satisfied (e.g., in our worst case configuration in Definition 3.4). However, under certain assumptions, it is possible to show ($\clubsuit$) > 0. In what follows, we provide lower bounds on ($\clubsuit$) assuming standard OCRS and random-element OCRS with L-partite hypergraphs, respectively.

4.1 Standard OCRS

In the standard OCRS problem, there exists at most one possible product in each time step, i.e., $|N_t| = 1$ for all t. With such a restriction, it is not possible to choose the products in such a way that $(\clubsuit) = 0$. In fact, we show in the following result that $(\clubsuit) \geq L - 1$.

Lemma 4.2. *Under standard OCRS, it holds that* $(\clubsuit) \geq L - 1$.

The proof of Lemma 4.2 appears in Appendix B, and so we just briefly sketch it here. Using the fact that the feasibility constraint (8) is binding, and that every product intersects with $[L]$ in at most one item, we can rephrase $(\clubsuit)$ as an optimization problem maximizing

$$\sum_{i=1}^{L} \sum_{i'' \in [N] \setminus [L]} \left(\sum_{j: i, i'' \in A_j} x_j \right) \left(\sum_{i' \neq i} \sum_{j': i', i'' \in A_{j'}} x_{j'} \right).$$

The problem can be further rewritten as an optimization problem with a bilinear objective and linear constraints. Interestingly, we are able to characterize the optimal solution, which leads to the lemma. By combining Lemma 4.2 with the derivation preceding (9), we get the following result:

Theorem 4.3. *Given $L \geq 2$, suppose that π of Definition 2.1 is passed α which satisfies*

$$\kappa(\alpha) := 1 - \alpha(L+1) + \alpha^2 \frac{L-1}{L} \left(\frac{1 - \alpha(L+1) + \alpha/2L}{1 - \alpha L + \alpha/2L} \right)^{2L} \geq 0.$$

Then, π is α-selectable on standard OCRS inputs.

By checking the first order derivative of κ, it can be verified that the function $\kappa(\alpha)$ is monotonically decreasing in α. Since $\kappa(1/(1+L)) > 0$, this implies that there exists $\alpha^* > 1/(1+L)$ such that $\kappa(\alpha^*) = 0$. Thus, π is α^*-selectable, and so $1/(L+1)$ is beatable. For any given L, we can numerically find the value of α^*. In particular, when $L = 2$, we have $\alpha^* \approx 0.33336$.

4.2 Random-element OCRS with an L-partite Graph

Theorem 4.3 shows that $1/(1+L)$ is beatable for any value of L under the standard OCRS, and so combined with Theorem 3.1, we have proven a separation between standard OCRS and random-element OCRS when L is a prime power. We now show that if the underlying graph has some structural properties, then $1/(L+1)$ is beatable even for random-element OCRS. We focus on the case where the products and items form an L-partite hypergraph. Specifically, the set of items can be partitioned into L disjoint subsets, such that every product contains at most one item from each subset.

Definition 4.1 (L-partite Hypergraph)**.** We say that the feasibility structure forms an *L-partite hypergraph* if the item set M can be partitioned into $M_1 \cup \cdots \cup M_L$ such that $|A_j \cap M_\ell| \leq 1$ for all products $j \in N$ and $\ell = 1, \ldots, L$. Put in words, the items can be divided into L groups such that each product contains at most one item from each group.

Hypergraphs of this form have been widely studied in NRM. For example, in the assemble-to-order system, all products are assembled from a set of components so that different combinations of items for each component lead to different products. Without loss of generality, we assume each product j is consists of L items with exactly one item from each M_i. If there exists a product which contains less than L items, we can add a dummy item to the group which is consumed by this product. We now argue that $1/(1+L)$ is beatable in this setting. As in the case of standard OCRS inputs, it suffices to lower bound ($\clubsuit$).

Lemma 4.4. *For an L-partite hypergraph, it holds that* ($\clubsuit$) ≥ 1.

The proof of Lemma 4.4 is analogous to Lemma 4.2 in that it involves characterizing the optimal solution to a related optimization problem. Combined with the previous discussion, Theorem 4.5 then follows.

Theorem 4.5. *Given $L \geq 2$, suppose that π of Definition 2.1 is passed α which satisfies*

$$1 - \alpha(L+1) + \frac{\alpha^2}{L}\left(\frac{1-\alpha(L+1)+\alpha/2L}{1-\alpha L+\alpha/2L}\right)^{2L} \geq 0.$$

Then, π is α-selectable on L-partite hypergraphs.

The left-hand side function of Theorem 4.5 is decreasing in α and greater than 0 at $\alpha = 1/(L+1)$. Thus, $1/(L+1)$ is beatable for L-partite hypergraphs.

5 Reduction

All of this paper was focused on deriving (random-element) OCRS's. In this section, we define applications in the form of the Network Revenue Management and Online Combinatorial Auctions problems, along with various special cases, and formalize their reduction to (random-element) OCRS's. We define a very general problem that, while abstract, allows us to unify the two applications and simultaneously derive $1/(1+L)$ (and better) guarantees for them, using OCRS. This very general abstraction, along with the distinction between standard vs. random-element OCRS in the reduction, is to our knowledge new.

Definition 5.1 (Abstract Problem with Substitutable Actions)**.** Items $i \in M$ have positive integer starting inventories k_i. Products $j \in N$ have fixed rewards $r_j \geq 0$ and require a non-empty subset of items $A_j \subseteq M$. At each time $t = 1, \dots, T$, an *action* $S \in \mathcal{S}_t$ is played, resulting in up to one product j being sold, in which case reward r_j is collected and the remaining inventory of each $i \in A_j$ is decremented by 1. A product j becomes *infeasible* if it requires an item with zero remaining inventory, and actions that have positive probability of selling an infeasible product cannot be played. The objective is to maximize total expected reward, when all sales probabilities are known in advance and independent across time. In particular, for all t and $S \in \mathcal{S}_t$, we are told the probability $\phi_t(j, S)$ of selling each product j under action S, where $\sum_j \phi_t(j, S) \leq 1$ (because at most one product can be sold) and $1 - \sum_j \phi_t(j, S)$ denotes the probability that no product is sold.

We assume that ϕ_t defines *substitutable actions* for all t. By this, we mean that for any action $S \in \mathcal{S}_t$ and set of "forbidden" products $F \subseteq N$, there exists a "recourse" action $S' \in \mathcal{S}_t$ such that

$$\phi_t(j, S') = 0 \qquad \forall j \in F; \tag{10}$$

$$\phi_t(j, S') \geq \phi_t(j, S) \qquad \forall j \notin F. \tag{11}$$

Put in words, the recourse action S' must have zero probability of selling any forbidden product, and weakly greater probability of selling any non-forbidden product. Taking $F = N$, condition (10) implies the existence of a "null" action in each $\mathcal{S}_t$ that has zero probability of selling any product.

The problem instance falls under the special case of *no substitution* if for each t, the set of products that can be sold under *any* action must all require the same subset of items (even though these products can have different rewards). Formally, this is stated as $A_j = A_{j'}$ (but possibly $r_j \neq r_{j'}$) for all $j, j' \in \cup_{S \in \mathcal{S}_t} \{j : \phi_t(j, S) > 0\}$, for each $t = 1, \ldots, T$.

Definition 5.2 (Capturing NRM)**.** The abstract problem directly defines NRM if actions are interpreted as *assortments* (subsets) of products to offer, i.e. $S \subseteq N$, with $\phi_t(j, S) = 0$ for all $j \notin S$. Function ϕ_t defines substitutable actions via the recourse action $S' = S \setminus F$, as long as the probability of selling products in an assortment does not decrease after other products F are removed (and $\mathcal{S}_t$ is downward-closed in that if S lies in $\mathcal{S}_t$ then all subsets of S also lie in $\mathcal{S}_t$). This holds for *substitutable* choice models, which is a standard assumption in assortment optimization that is satisfied by all random-utility models (Golrezaei et al., 2014). In our setting that sells products which are *bundles* of items, we argue that this assumption is even milder, because complementarity effects can be captured by creating larger bundles that combine all the items that are complements.

We note that the formulation with assortments is general, and captures pricing decisions as well. Indeed, one can make copies of each product j, where the copies have identical A_j but different r_j, and make $\mathcal{S}_t$ constrain assortment S so that at most one copy (price) of each product is offered.

In the accept/reject version of NRM, at each time t a random product j arrives, drawn independently according to a known probability vector $(\lambda_{tj})_j$. This can be captured using assortments by defining $\phi_t(j, S) = \lambda_{tj} \mathbb{1}(j \in S)$ for all t, j, S, with S representing the subset of products to make available at time t. Although this is often called the "independent demand model" in the literature, under our Definition 5.1 it is not a case of no substitution, because products that require different sets of items can all have positive probability of arriving at a time step t. Put another way, one product arriving during t precludes other products from arriving, inducing a basic form of negative correlation. However, the original formulation of NRM (Gallego and Van Ryzin, 1997), in which time is continuous and demands for different products arrive from independent (time-varying) Poisson processes, falls under the special case of no substitution because the time steps are infinitesimally small and any negative correlation will vanish.

Definition 5.2 as stated does not capture *personalized* revenue management, in which a customer type is observed at each time t before assortment S is decided. Nonetheless, personalized NRM can be captured using our abstract Definition 5.1, by having an action represent a *mapping* that prescribes a decision for each customer type that could be observed. We now illustrate this, by capturing similar dynamics in the OCA problem, in which for each t, a type (valuation function) is observed before a decision is made.

Definition 5.3 (Capturing L-bounded Online Combinatorial Auctions)**.** In the Online Combinatorial Auctions problem, each t represents an agent, who independently draws a random valuation function $V_t : 2^M \to \mathbb{R}_{\geq 0}$ from a known distribution. It is assumed that every potential realization of V_t satisfies $V_t(\emptyset) = 0$, monotonicity ($A' \subseteq A \implies V_t(A') \leq V_t(A)$), and $V_t(A) = \max_{A' \subseteq A, |A'|=L} V_t(A')$ for all $|A| > L$, where the last assumption is the critical one capturing the fact that an agent never needs more than L items. When an agent t arrives, V_t is observed, and then a subset of at most L items must be irrevocably assigned to them, subject to the same inventory constraints as in Definition 5.1. The objective is to maximize expected welfare, i.e. the expected sum of valuations that agents have for the items assigned to them. We do not worry about incentive-compatibility, although recent developments (Banihashem et al., 2024) show that our algorithm can be converted into an incentive-compatible posted-price mechanism.

To capture this using the abstract problem in Definition 5.1, for each t, potential realization of V_t, and bundle $A \subseteq M$ with $1 \leq |A| \leq L$, we create a product j with $A_j = A$ and $r_j = V_t(A)$. An action S is a mapping that assigns items for each potential realization of V_t, among the $2^L - 1$ products created for that realization, or the empty set which is not a product. For products j, probability $\phi_t(j, S)$ equals that of realizing V_t if j is assigned for V_t by S, and 0 otherwise. (We worry about the computational efficiency of these operations later.) This defines substitutable actions because for any mapping S, we can take S' to be the mapping that remaps any forbidden products F in the range of S to the empty set, satisfying (10) by construction, and satisfying (11) as equality.

In the *single-minded* special case, each agent t is only interested in a particular non-empty bundle $A^t \subseteq M$. That is, $V_t(A) = V_t(A^t)$ if $A \supseteq A^t$ and $V_t(A) = 0$ otherwise. The only uncertainty lies in the valuation $V_t(A^t)$, and hence this can also be interpreted as a *prophet inequality* problem. Indeed, we only have to create products j with $A_j = A^t$ for each t, and an action $S \in \mathcal{S}_t$ would decide for every potential realization of $V_t(A^t)$ whether it is high enough to "accept" by assigning A^t. Returning to the abstract problem, this would fall under the no substitution special case.

We now define a relaxation for the abstract problem that will allow us to derive guarantees for the NRM and OCA problems in a unified manner.

Definition 5.4 (LP Relaxation)**.** Let LP denote the optimal objective value of the following LP:

$$\mathsf{LP} := \max \sum_j r_j \sum_{t=1}^{T} \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S) \tag{12}$$

$$\text{s.t.} \sum_{j: i \in A_j} \sum_{t=1}^{T} \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S) \le k_i \qquad \forall i \in M \tag{13}$$

$$\sum_{S \in \mathcal{S}_t} x_t(S) = 1 \qquad \forall t = 1, \ldots, T \tag{14}$$

$$x_t(S) \ge 0 \qquad \forall t = 1, \ldots, T; S \in \mathcal{S}_t. \tag{15}$$

In (12)–(15), variable $x_t(S)$ can be interpreted as the probability of playing action S at time t. We note that the item feasibility constraints only have to be satisfied in expectation in (13). The optimal objective value LP is an upper bound on the expected welfare of the prophet in OCA, who knows the realizations of V_t in advance and assigns items optimally. For the special case of the accept-reject NRM problem, LP can be seen to be equivalent to the fluid LP (i.e, (5)) from Section 3, and so it upper bounds the expected reward of the optimal offline allocation. When assortments are offered in the general NRM problem (Ma, 2022), there is no clear analogue of this benchmark, but LP still upper-bounds the optimal (intractable) dynamic programming value, which is well-defined assuming the time steps unfold in chronological order $t = 1, \ldots, T$.

Theorem 5.1. *For the abstract problem with substitutable actions, a random-element α-selectable OCRS implies an online algorithm whose total expected reward is at least $\alpha \cdot \mathsf{LP}$. If the instance has no substitution, then a standard OCRS (without random elements) suffices.*

Taken abstractly, Theorem 5.1 does not promise anything about computational efficiency. However, we will see during its proof that for both the NRM and OCA problems, our OCRS's (which are polynomial-time) will imply polynomial-time online algorithms. Theorem 5.1 allows us to achieve the guarantee of $1/(1+L)$ in both the general NRM and OCA problems, and beat $1/(1+L)$ in the independent Poisson demand and single-minded special cases, respectively. We can also always beat $1/(1+L)$ if the products form an L-partite hypergraph (see Definition 4.1), and we now clarify how this arises from a further special case of valuation functions.

Definition 5.5 (L-partite Valuation Function)**.** Recall that a valuation function $V : 2^M \to \mathbb{R}_{\geq 0}$ is L-bounded if $V(A) = \max_{A' \subseteq A, |A'|=L} V(A')$ whenever $|A| > L$, which we assumed about the agents' valuation functions. We define an *L-partite valuation function* as the further special case where

$$V(A) = \max_{i_1 \in A \cap M_1, \ldots, i_L \in A \cap M_L} V(\{i_1, \ldots, i_L\}). \tag{16}$$

Here we assume that the items M are pre-divided into L groups $M_1, \ldots, M_L$, and note that i_ℓ may not exist in (16) if $A \cap M_\ell = \emptyset$. Put in words, (16) imposes that any subset A is valued based on the maximum valuation obtainable by choosing at most one item from each group within A. When reducing from L-partite valuation functions to the abstract problem, we only have to create products j where A_j satisfies $|A_j \cap M_\ell| \leq 1$ for all $\ell = 1, \ldots, L$, and hence the products will form an L-partite hypergraph.

In Definitions 5.3 and 5.5 there were items and valuation functions but no products. We explained how to construct products for our abstract problem in Definition 5.1, in a way that translated L-bounded valuation functions to L-bounded products, and L-partite valuation functions to L-partite hypergraphs. In the next setting we capture, there are products and feasibility constraints but no items (or valuation functions). We explain how to construct items, starting inventories, and item containment relationships that represent the same feasibility constraints and correspond to an L-partite hypergraph.

Definition 5.6 (Intersection of L Partition Matroids)**.** In a partition matroid constraint, a universe of products N is partitioned into parts $N(1), \ldots, N(m)$, with upper bounds $k(1), \ldots, k(m)$. A subset $S \subseteq N$ is said to be feasible if $|S \cap N(i)| \leq k(i)$ for all $i = 1, \ldots, m$. Given L partition matroids defined by parts $N^\ell(1), \ldots, N^\ell(m_\ell)$ and upper bounds $k^\ell(1), \ldots, k^\ell(m_\ell)$ for $\ell = 1, \ldots, L$, their intersection refers to subsets $S \subseteq N$ that are feasible in each matroid ℓ.

We can translate the intersection of L partition matroids into inventory constraints that form an L-partite hypergraph, as follows. For each partition matroid $\ell = 1, \ldots, L$, we create a group of items M_ℓ, with one item for each $i = 1, \ldots, m_\ell$ whose starting inventory is $k^\ell(i)$. Each product $j \in N$ then requires from each group ℓ the item $i \in \{1, \ldots, m_\ell\}$ for which $j \in N^\ell(i)$. Defining A_j like this for all $j \in N$, it is direct to check that these products form an L-partite hypergraph.

5.1 Algorithm and Proof for Theorem 5.1

Our algorithm has two initial processing steps. First it solves the LP relaxation (12)–(15), hereafter letting $x_t(S)$ denote the values in an optimal solution. Although the LP as written could has exponentially many variables due to the size of $\mathcal{S}_t$, its dual has a separation oracle as long as for any t and weights $\{r'_j : j \in N\}$, one can efficiently solve the optimization problem

$$\max_{S \in \mathcal{S}_t} \sum_j r'_j \phi_t(j, S). \tag{17}$$

(17) is trivially solved in OCA, because the optimal S would map each potential realization of V_t to its corresponding product j with the maximum r'_j, or no product if all weights are negative. (17) also coincides exactly with the single-shot assortment optimization problem in NRM, which can be solved for commonly-used choice models, leading to a separation oracle (Gallego et al., 2004). By the equivalence of separation and optimization (Korte et al., 2011), tractability of (17) implies that the LP relaxation can be solved in polynomial time.

The second initial processing step is to duplicate items and products to transform to an instance where the items M all have an initial inventory of 1, and the products N are partitioned into $N_1 \cup \cdots \cup N_T$ such that $\phi_t(j, S) > 0$ only if $j \in N_t$. This would allow us to define $x_j := \sum_{S \in \mathcal{S}_t} \phi_t(j, S) x_t(S)$ for all $t = 1, \ldots, T, j \in N_t$ and satisfy the conditions of random-element OCRS, noting that $x_j \leq 1$ must hold if initial inventories are 1. Moreover, if the original problem instance had no substitution, then we would want $A_j = A_{j'}$ for any $j, j' \in N_t$, for all t, in the transformed instance. This is equivalent to the condition of $|N_t| = 1$ for all t and allows us to apply a standard OCRS, where the equivalence is because an OCRS does not discriminate products based on r_j. In Appendix C.1, we describe a transformation that satisfies all of these properties.

Having completed the initial processing, our online algorithm is to, for each t:

1. Query the OCRS to obtain a random bit vector $(B_j)_{j \in N_t}$, where $B_j \in \{0, 1\}$ indicates whether the OCRS *would* accept each product $j \in N_t$ if it were to be the active product for t;
2. Play a (randomized) action from $\mathcal{S}_t$ such that the probability of selling each product $j \in N_t$ is x_j if $B_j = 1$, and 0 if $B_j = 0$.

The OCRS guarantees $\mathbb{E}[B_j] = \alpha$ for all j, which would imply that every product $j \in N_t$ gets sold w.p. αx_j. This argument requires the independence of sales across time, because B_j at the current time t depends on the inventory state, which in turn depends on the sales realizations before t. Given this, the online algorithm has total expected reward $\sum_{t=1}^{T} \sum_{j \in N_t} \alpha r_j x_j$, which equals $\alpha \cdot \mathsf{LP}$ as claimed in Theorem 5.1. We formally prove the validity of this online algorithm and the OCRS guarantee in Appendix C.2, which also requires the following lemma for substitutable actions.

Lemma 5.2. *Suppose that ϕ_t defines substitutable actions for selling products in N_t using actions in $\mathcal{S}_t$. Then for all $S \in \mathcal{S}_t$ and $F \subseteq N_t$, one can compute a randomized S' such that*

$$\begin{aligned} &\mathbb{E}_{S'}[\phi_t(j, S')] = 0 && \forall j \in F; \\ &\mathbb{E}_{S'}[\phi_t(j, S')] = \phi_t(j, S) && \forall j \notin F. \end{aligned} \tag{18}$$

(18) differs from the original condition (11) for substitutable actions by saying that we can sell each non-forbidden product w.p. *exactly* $\phi_t(j, S)$, after averaging over a random recourse action S'. This is important for OCRS's, because selling non-forbidden products w.p. higher than originally prescribed may cause other products to become infeasible with too high probability. Lemma 5.2 is not necessary for the OCA problem, as noted earlier, because the recourse action S' by definition will satisfy (11) as equality.

Results similar to Lemma 5.2 have appeared in various revenue management papers where the action is to offer an assortment. The need for such a result arises in revenue management with reusable resources, in which it has been called "sub-assortment sampling" (Feng et al., 2022) and "probability match" (Goyal et al., 2020). A similar result was used earlier to ensure that items are not sold with probability higher than intended in Chen et al. (2023), in which it was called random assortment from "breakpoints". These results are proved based on the following idea—if (11) is satisfied as strict inequality for some products, then one can add the greatest violator to F with some probability to scale down its selling probability, and repeat until (11) is satisfied as equality for all products. In doing so, one generates a sequence of breakpoints that defines a randomized F, which induces a randomized sub-assortment S', ultimately matching the original probabilities $\phi_t(j, S)$ for all $j \notin F$. We provide a self-contained proof of Lemma 5.2 in Appendix C.3, and this completes our reduction.

6 Conclusion and Open Questions

We recap the main contributions of this paper. First, we beat the benchmark of $1/(1+L)$ that has appeared in many papers about Network Revenue Management or Online Combinatorial Auctions. Also, we demonstrate that the subtlety of whether elements are random can affect the best-possible guarantees in OCRS. Finally, we define an extended notion of "random-element" OCRS that is necessary to handle the general NRM and OCA problems in a black-box manner.

We end by posing a few open questions. First, one could also distinguish between standard vs. random-element OCRS under other arrival orders, e.g. random order. There, a natural benchmark (the equivalent of $1/(1+L)$) would be $(1-\exp(-L))/L$, as established in the IID setting by Marinkovic et al. (2023). Second, our analysis does not naturally lend itself to improved guarantees if all items have large initial inventories. It may be interesting to interpolate between our guarantees and Amil et al. (2023), whose guarantees for NRM do improve with large inventories. Finally, our counterexample has the curious property of relying on a finite affine plane of order L. Might it be possible to beat $1/7$ for random-element OCRS when $L = 6$?

Acknowledgements. The authors thank Huseyin Topaloglu for insightful early discussions.

References

Ayoub Amil, Ali Makhdoumi, and Yehua Wei. 2023. Multi-item order fulfillment revisited: Lp formulation and prophet inequality. *EC* (2023).

Jackie Baek and Will Ma. 2022. Bifurcating constraints to improve approximation ratios for network revenue management with reusable resources. *Operations Research* 70, 4 (2022), 2226–2236.

Kiarash Banihashem, MohammadTaghi Hajiaghayi, Dariusz Kowalski, Piotr Krysta, and Jan Olkowski. 2024. *Power of Posted-price Mechanisms for Prophet Inequalities.* 4580–4604. https://doi.org/10.1137/1.9781611977912.163

Xi Chen, Will Ma, David Simchi-Levi, and Linwei Xin. 2023. Assortment planning for recommendations at checkout under inventory constraints. *Mathematics of Operations Research* (2023).

José Correa, Andrés Cristi, Andrés Fielbaum, Tristan Pollner, and S Matthew Weinberg. 2023. Optimal item pricing in online combinatorial auctions. *Mathematical Programming* (2023), 1–32.

Shaddin Dughmi. 2020. The Outer Limits of Contention Resolution on Matroids and Connections to the Secretary Problem. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Paul Dutting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. 2020. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.* 49, 3 (2020), 540–582.

Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. 2022. Prophet Matching with General Arrivals. *Mathematics of Operations Research* 47, 2 (2022), 878–898.

Yiding Feng, Rad Niazadeh, and Amin Saberi. 2022. Near-optimal bayesian online assortment of reusable resources. In *Proceedings of the 23rd ACM Conference on Economics and Computation.* 964–965.

Guillermo Gallego, Garud Iyengar, Robert Phillips, and Abhay Dubey. 2004. Managing flexible products on a network. *Available at SSRN 3567371* (2004), .

Guillermo Gallego and Garrett Van Ryzin. 1997. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations research* 45, 1 (1997), 24–41.

Negin Golrezaei, Hamid Nazerzadeh, and Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* 60, 6 (2014), 1532–1551.

Vineet Goyal, Garud Iyengar, and Rajan Udwani. 2020. Asymptotically optimal competitive ratio for online allocation of reusable resources. *arXiv preprint arXiv:2002.02430* (2020).

Jiashuo Jiang, Will Ma, and Jiawei Zhang. 2022. Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1221–1246.

Robert Kleinberg and Seth Matthew Weinberg. 2012. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 123–136.

Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. 2011. *Combinatorial optimization*. Vol. 1. Springer.

Euiwoong Lee and Sahil Singla. 2018. Optimal online contention resolution schemes via ex-ante prophet inequalities. In *26th European Symposium on Algorithms, ESA 2018*. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, .

Will Ma. 2022. Revenue-optimal deterministic auctions for multiple buyers with ordinal preferences over fixed-price items. *ACM Transactions on Economics and Computation* 10, 2 (2022), 1–32.

Yuhang Ma, Paat Rusmevichientong, Mika Sumida, and Huseyin Topaloglu. 2020. An approximation algorithm for network revenue management under nonstationary arrivals. *Operations Research* 68, 3 (2020), 834–855.

Calum MacRury, Will Ma, and Nathaniel Grammel. 2023. On (Random-order) Online Contention Resolution Schemes for the Matching Polytope of (Bipartite) Graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1995–2014.

Javier Marinkovic, José A Soto, and Victor Verdugo. 2023. Online Combinatorial Assignment in Independence Systems. *arXiv preprint arXiv:2311.00890* (2023).

G Eric Moorhouse. 2007. Incidence geometry. *University of Wyoming* (2007). `http://ericmoorhouse.org/handouts/Incidence_Geometry.pdf`

Yosef Rinott and Ester Samuel-Cahn. 1987. Comparisons of optimal stopping values and prophet inequalities for negatively dependent random variables. *The Annals of Statistics* 15, 4 (1987), 1482–1490.

Paat Rusmevichientong, Mika Sumida, Huseyin Topaloglu, and Yicheng Bai. 2023. Revenue Management with Heterogeneous Resources: Unit Resource Capacities, Advance Bookings, and Itineraries over Time Intervals. *Operations Research* (2023).

Ester Samuel-Cahn. 1991. Prophet inequalities for bounded negatively dependent random variables. *Statistics & probability letters* 12, 3 (1991), 213–216.

Wenchang Zhu and Huseyin Topaloglu. 2023. Performance guarantees for network revenue management with flexible products. *Manufacturing & Service Operations Management* (2023).

A Supplement to Section 2

The policy π defined in Definition 2.1 cannot be implemented directly because it requires the knowledge of the probability $\mathbb{P}(F_j)$ for every product j. In what follows, we provide a policy with the aid of simulation so that it can be implemented.

Definition A.1. For each time t, run a Monte Carlo simulation with K trails: starting from time $\tau = 1$ to $\tau = t-1$, implement the policy π in Definition 2.1 with $\hat{\mathbb{P}}(F_j)$ for $j \in N_1 \cup \cdots \cup N_{t-1}$ and set $\alpha = (1-\varepsilon)/(1+L)$. Let $\hat{\mathbb{P}}(F_j)$ denote the empirical estimation of the probability that the product $j \in N_t$ is available, that is,

$$\hat{\mathbb{P}}(F_j) = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}\{\text{product } j \text{ is available in } k\text{-th trial}\}.$$

Let $\hat{\pi}$ denote the simulation algorithm and $\hat{\mathbb{P}}(F_j)$ denote the output of the simulation algorithm. Moreover, let $\mathbb{P}^{\hat{\pi}}(F_j)$ denote the true probability that product j is available under policy $\hat{\pi}$, which is a random variable depending on the previous sample paths. Note that by construction, $\hat{\mathbb{P}}(F_j)$ is an unbiased estimate of $\mathbb{P}^{\hat{\pi}}(F_j)$. Let $V^{\hat{\pi}}$ denote the expected rewards of the simulation based policy $\hat{\pi}$.

Lemma A.1. *For any time t, given that $\alpha = (1-\varepsilon)/(1+L)$ and $\mathbb{P}^{\hat{\pi}}(F_j)/\hat{\mathbb{P}}(F_j) \le 1/(1-\varepsilon)$ for all $\tau < t$ and $j \in N_\tau$, it holds that $\mathbb{P}^{\hat{\pi}}(F_j) \ge 1/(1+L)$ for any $j \in N_t$.*

Proof of Lemma A.1. Note that for any $j \in N_t$,

$$\begin{aligned}
\mathbb{P}^{\hat{\pi}}(F_j) =& \mathbb{P}^{\hat{\pi}}\left(\cap_{i \in A_j} F_{ti}\right) = 1 - \mathbb{P}^{\hat{\pi}}\left(\cup_{i \in A_j} \bar{F}_{ti}\right) \ge 1 - \sum_{i \in A_j} \mathbb{P}^{\hat{\pi}}\left(\bar{F}_{ti}\right) \\
=& 1 - \sum_{i \in A_j} \frac{1-\varepsilon}{1+L} \sum_{\tau=1}^{t-1} \sum_{j' \in N_\tau : i \in A_{j'}} x_{j'} \cdot \frac{1}{\hat{\mathbb{P}}(F_{j'})} \cdot \mathbb{P}^{\hat{\pi}}\left(F_{j'}\right) \\
\ge& 1 - \sum_{i \in A_j} \frac{1-\varepsilon}{1+L} \sum_{\tau=1}^{t-1} \frac{1}{1-\varepsilon} \sum_{j' \in N_\tau : i \in A_{j'}} x_{j'} \\
\ge& 1 - \frac{1}{1+L} |A_j| \ge \frac{1}{1+L}.
\end{aligned}$$

where the first inequality holds due to the assumption. □

Theorem A.2. *For any $\varepsilon \in (0,1)$, by taking $K = \frac{3(1+L)}{\varepsilon^2}\log\left(\frac{2TM}{\varepsilon}\right)$, it holds that $V^{\hat{\pi}} \geq \frac{(1-\varepsilon)^2}{1+\varepsilon}\frac{1}{1+L}V^*$.*

Proof of Theorem A.2. By union bound and Bayes rule, we have

$$
\begin{aligned}
&\mathbb{P}\left(\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}, \forall j\right)\\
=&\mathbb{P}\left(\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall j\right)\\
=&\prod_{t=1}^{T}\mathbb{P}\left(\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall j \in N_t\middle|\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall(\tau < t, j \in N_\tau)\right)\\
\geq&1 - \sum_{t=1}^{T}\mathbb{P}\left(\exists j \in N_t, \left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| > \varepsilon\mathbb{P}^{\hat{\pi}}(F_j)\middle|\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall(\tau < t, j \in N_\tau)\right)\\
\geq&1 - \sum_{t=1}^{T}\sum_{j\in N_t}\mathbb{P}\left(\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| > \varepsilon\mathbb{P}^{\hat{\pi}}(F_j)\middle|\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall(\tau < t, j \in N_\tau)\right)\\
\overset{(a)}{\geq}&1 - \sum_{t=1}^{T}\sum_{j\in N_t}2\mathbb{E}^{\hat{\pi}}\left[\exp\left(-\frac{K}{3}\varepsilon^2\mathbb{P}^{\hat{\pi}}(F_j)\right)\middle|\left|\mathbb{P}^{\hat{\pi}}(F_j) - \hat{\mathbb{P}}(F_j)\right| \leq \varepsilon\mathbb{P}^{\hat{\pi}}(F_j), \forall(\tau < t, j \in N_\tau)\right]\\
\overset{(b)}{\geq}&1 - 2TM\exp\left(-\frac{\varepsilon^2 K}{3(1+L)}\right),
\end{aligned}
$$

where inequality (a) follows from Chernoff bound and inequality (b) follows from Lemma A.1. Therefore, by taking

$$K = \frac{3(1+L)}{\varepsilon^2}\log\left(\frac{2TM}{\varepsilon}\right),$$

we have

$$\mathbb{P}\left(\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}, \forall j\right) \geq 1-\varepsilon.$$

Thus, we have

$$
\begin{aligned}
V^{\hat{\pi}} =& \mathbb{E}^{\hat{\pi}}\left[\sum_{t=1}^{T}\sum_{j=1}^{M} r_j Z_j\right] = \sum_{t=1}^{T}\sum_{j\in N_t} r_j \mathbb{E}^{\hat{\pi}}\left[Z_j\right]\\
\geq&\sum_{t=1}^{T}\sum_{j\in N_t} r_j\mathbb{P}\left(\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}\right)\mathbb{E}^{\hat{\pi}}\left[Z_j\middle|\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}\right]\\
=&\alpha\sum_{t=1}^{T}\sum_{j\in N_t} r_j x_j\mathbb{P}\left(\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}\right)\mathbb{E}^{\hat{\pi}}\left[\frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)}\middle|\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}\right]\\
\geq&\frac{1-\varepsilon}{(1+\varepsilon)(1+L)}\sum_{t=1}^{T}\sum_{j\in N_t} r_j x_j\mathbb{P}\left(\frac{1}{1+\varepsilon} \leq \frac{\mathbb{P}^{\hat{\pi}}(F_j)}{\hat{\mathbb{P}}(F_j)} \leq \frac{1}{1-\varepsilon}\right)
\end{aligned}
$$

$$\geq \frac{(1-\varepsilon)^2}{(1+\varepsilon)} \frac{V^*}{1+L}.$$

□

B Supplement to Section 4

Proof of Lemma 4.1. Since each element is consisted of at most L items and $A_j \cap A_{j'} = \emptyset$, we have $\left|A_j \cup A_{j'}\right| \leq 2L$. For simplicity, let $\mathcal{L} = \left|A_j \cup A_{j'}\right| \geq 2$ and assume the $\mathcal{L}$ items are indexed by $\{1, \ldots, \mathcal{L}\}$ without loss of generality. We can then partition the set $\{j'' : A_{j''} \cap (A_j \cup A_{j'}) \neq \emptyset\}$ into $\mathcal{L}$ disjoint sets $(J_i)_{i=1}^{\mathcal{L}}$ such that

$$J_i \subseteq \{j : i \in A_j\}, \sum_{j \in J_i} x_j \leq 1, \ \forall i = 1, \ldots, \mathcal{L}.$$

Moreover, for any product $j \in J_i$, let t be the time such that $j \in N_t$, recall that Z_j denote the event that product j is accepted and we have $\mathbb{P}(Z_j) = \alpha x_j$. Therefore, it holds that

$$\begin{aligned}
&\mathbb{P}\left(F_j\right) = \mathbb{P}\left(\cap_{i' \in A_j} F_{ti'}\right) \geq 1 - \sum_{i' \in A_j} \mathbb{P}(\bar{F}_{ti'}) \\
&\geq 1 - \sum_{i' \in A_j} \sum_{\tau < t} \sum_{j' \in N_\tau : i' \in A_{j'}} \mathbb{P}(Z_{j'}) \geq 1 - \sum_{i' \in A_j} \sum_{t=1}^{T} \sum_{j' \in N_t : i' \in A_{j'}} \mathbb{P}(Z_{j'}) + \sum_{j' \in N_\tau : i \in A_{j'}} \mathbb{P}(Z_{j'}) \\
&\geq 1 - \alpha L + \alpha \sum_{j' \in N_t : i \in A_{j'}} x_{j'} \geq 1 - \alpha L + \alpha \sum_{j' \in N_t \cap J_i} x_{j'},
\end{aligned}$$

where the second inequality holds because the item i' is available at time t only if no associated product j' has been accepted before, and the last inequality holds because $J_i \subseteq \{j : i \in A_j\}$. The inequality above implies that

$$\begin{aligned}
&\prod_{\tau=1}^{T} \left(1 - \sum_{j'' \in N_\tau : A_{j''} \cap \left(A_j \cup A_{j'}\right) \neq \emptyset} \frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})}\right) \\
=&\prod_{\tau=1}^{T} \left(1 - \sum_{i=1}^{\mathcal{L}} \sum_{j'' \in N_\tau \cap J_i} \frac{\alpha x_{j''}}{\mathbb{P}(F_{j''})}\right)
\end{aligned}$$

$$\geq \prod_{\tau=1}^{T}\left(1-\sum_{i=1}^{\mathcal{L}}\sum_{j''\in N_\tau\cap J_i}\frac{\alpha x_{j''}}{1-\alpha L+\alpha\sum\limits_{j'\in N_\tau\cap J_i}x_{j'}}\right)$$
$$=\prod_{\tau=1}^{T}\left(1-\sum_{i=1}^{\mathcal{L}}\frac{\alpha\sum\limits_{j'\in N_\tau\cap J_i}x_{j'}}{1-\alpha L+\alpha\sum\limits_{j'\in N_\tau\cap J_i}x_{j'}}\right).$$

To provide a lower bound to the term above, we consider an optimization problem. Note that

$$\sum_{i=1}^{\mathcal{L}}\sum_{j'\in N_\tau\cap J_i}x_{j'}\leq\sum_{j'\in N_\tau}x_{j'}\leq 1,\forall\tau.$$

Moreover, by the feasibility constraint, it also holds that

$$\sum_{\tau=1}^{T}\sum_{j'\in N_\tau\cap J_i}x_{j'}\leq\sum_{\tau=1}^{T}\sum_{j'\in N_\tau:i\in A_{j'}}x_{j'}=\sum_{j':i\in A_{j'}}x_{j'}\leq 1,\forall i.$$

Let $y_{\tau i}=\sum_{j'\in N_\tau\cap J_i}x_{j'}$, it is sufficient to the optimization problem as follows:

$$\min_{y_{\tau i}\geq 0}\prod_{\tau=1}^{T}\left(1-\sum_{i=1}^{\mathcal{L}}\frac{\alpha y_{\tau i}}{1-\alpha L+\alpha y_{\tau i}}\right)\ \text{s.t.}\ \sum_{i=1}^{\mathcal{L}}y_{\tau i}\leq 1,\ \forall\tau,\ \sum_{\tau=1}^{T}y_{\tau i}\leq 1,\ \forall i,$$
$$=\exp\left(\min_{y_{\tau i}\geq 0}\sum_{\tau=1}^{T}\log\left(1-\sum_{i=1}^{\mathcal{L}}\frac{\alpha y_{\tau i}}{1-\alpha L+\alpha y_{\tau i}}\right)\right)\ \text{s.t.}\ \sum_{i=1}^{\mathcal{L}}y_{\tau i}\leq 1,\ \forall\tau,\ \sum_{\tau=1}^{T}y_{\tau i}\leq 1,\ \forall i,$$
$$\geq\exp\left(\min_{y_{\tau i}\geq 0}\sum_{\tau=1}^{T}\log\left(1-\sum_{i=1}^{\mathcal{L}}\frac{\alpha y_{\tau i}}{1-\alpha L+\alpha y_{\tau i}}\right)\right)\ \text{s.t.}\ \sum_{i=1}^{\mathcal{L}}y_{\tau i}\leq 1,\ \forall\tau,\ \sum_{\tau=1}^{T}\sum_{i=1}^{\mathcal{L}}y_{\tau i}\leq\mathcal{L}.$$

Note that fix a time τ, it is optimal to set $(y_{\tau i})_{i=1}^{\mathcal{L}}$ equally because the function $\sum\limits_{i=1}^{\mathcal{L}}\frac{\alpha y_{\tau i}}{1-\alpha L+\alpha y_{\tau i}}$ is concave in $(y_{\tau i})_{i=1}^{\mathcal{L}}$. To see this, it is sufficient to show that function $g(x)=\frac{\alpha x}{1-\alpha L+\alpha x}$ is concave. Note that α is chosen so that $1-\alpha L>0$, thus $g(x)=1-\frac{1-\alpha L}{1-\alpha L+\alpha x}$, which is concave. Therefore, the optimization problem above can be further lower bounded by

$$\exp\left(\min_{y_\tau\geq 0}\sum_{\tau=1}^{T}\log\left(1-\frac{\alpha y_\tau}{1-\alpha L+\alpha y_\tau/\mathcal{L}}\right)\right)\ \text{s.t.}\ y_\tau\leq 1,\ \forall\tau,\ \sum_{\tau=1}^{T}y_\tau\leq\mathcal{L}. \tag{19}$$

We claim that $f(x) = \log\left(1 - \frac{\alpha x}{1-\alpha L+\alpha x/\mathcal{L}}\right)$ is also a concave function when $0 \le \alpha \le 1/L$. Note that

$$f'(x) = \frac{1}{1 - \frac{\alpha x}{1-\alpha L+\alpha x/\mathcal{L}}} \frac{-\alpha(1-\alpha L + \alpha x/\mathcal{L}) + \alpha^2 x/\mathcal{L}}{(1-\alpha L+\alpha x/\mathcal{L})^2} = \frac{-\alpha(1-\alpha L)}{(1-\alpha L+\alpha x/\mathcal{L})(1-\alpha L-(\mathcal{L}-1)\alpha x/\mathcal{L})},$$

$$f''(x) = \frac{\alpha^2(1-\alpha L)}{\mathcal{L}(1-\alpha L+\alpha x/\mathcal{L})^2(1-\alpha L-(\mathcal{L}-1)\alpha x/\mathcal{L})^2}\left((\mathcal{L}-2)(\alpha L-1) - \frac{2(\mathcal{L}-1)\alpha x}{\mathcal{L}}\right) \le 0.$$

Therefore, the optimization problem (19) is to minimize a concave function with linear constraints, thus the optimal solution is obtained at an extreme point of the feasible region. Moreover, the coefficient matrix is totally unimodular, therefore, all vertices are integral. If $T \le \mathcal{L}$, the optimal value is $\left(1 - \frac{\alpha}{1-\alpha L+\alpha/\mathcal{L}}\right)^T \ge \left(1 - \frac{\alpha}{1-\alpha L+\alpha/\mathcal{L}}\right)^{\mathcal{L}}$, otherwise, the optimal value is $\left(1 - \frac{\alpha}{1-\alpha L+\alpha/\mathcal{L}}\right)^{\mathcal{L}}$. Therefore, we can conclude that a lower bound to Problem (19) is $\left(1 - \frac{\alpha}{1-\alpha L+\alpha/\mathcal{L}}\right)^{\mathcal{L}}$. Moreover, this bound is decreasing in $\mathcal{L}$ and we have $\mathcal{L} \le 2L$, thus the result follows. $\square$

Proof of Lemma 4.2. Recall that in the worst case, all products intersect with the set of items $\{1, \ldots, L\}$ at most once, then we have

$$\begin{aligned}
&\sum_{i=1}^{L}\sum_{i'\neq i}\sum_{t=1}^{T}\sum_{t'\neq t}\sum_{\substack{j\in N_t, j'\in N_{t'}:\\ A_j\cap A_{j'}=\emptyset\\ i\in A_j, i'\in A_{j'}}} x_j x_{j'} = \sum_{i=1}^{L}\sum_{i'\neq i}\sum_{\substack{(j,j'):\\ A_j\cap A_{j'}=\emptyset\\ i\in A_j, i'\in A_{j'}}} x_j x_{j'} \\
=&\sum_{i=1}^{L}\sum_{i'\neq i}\sum_{j:i\in A_j}\sum_{j':i'\in A_{j'}} x_j x_{j'}\left(1 - \mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\}\right) \\
=&\sum_{i=1}^{L}\sum_{i'\neq i}\left(\sum_{j:i\in A_j} x_j\right)\left(\sum_{j':i'\in A_{j'}} x_{j'}\right) - \sum_{i=1}^{L}\sum_{i'\neq i}\sum_{j:i\in A_j}\sum_{j':i'\in A_{j'}} x_j x_{j'}\mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\} \\
=&L(L-1) - \sum_{i=1}^{L}\sum_{i'\neq i}\sum_{j:i\in A_j}\sum_{j':i'\in A_{j'}} x_j x_{j'}\mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\} \\
\ge&L(L-1) - \sum_{i=1}^{L}\sum_{i'\neq i}\sum_{j:i\in A_j}\sum_{i''\in A_j\setminus\{i\}}\sum_{j':i',i''\in A_{j'}} x_j x_{j'} \\
=&L(L-1) - \underbrace{\sum_{i=1}^{L}\sum_{i''\in[N]\setminus\{1,\ldots,L\}}\left(\sum_{j:i,i''\in A_j} x_j\right)\left(\sum_{i'\neq i}\sum_{j':i',i''\in A_{j'}} x_{j'}\right)}_{(a)}.
\end{aligned}$$

where the first equality holds because $|N_t| = 1$ for any t in the standard OCRS model and the last equality holds because if $i \in A_j$ for $i \in \{1, \dots, L\}$, then $A_j \backslash \{i\} \cap \{1, \dots, L\} = \emptyset$. In order to upper bound term (a), for simplicity, let

$$\alpha_{ii''} = \sum_{j: i, i'' \in A_j} x_j, \ \beta_{ii''} = \sum_{i' \neq i} \sum_{j': i', i'' \in A_{j'}} x_{j'}.$$

Note that for any fixed $i \in \{1, \dots, L\}$, it holds that

$$\alpha_{ii''} + \beta_{ii''} = \sum_{j: i, i'' \in A_j} x_j + \sum_{i' \neq i} \sum_{j: i', i'' \in A_{j'}} x_{j'} = \sum_{i=1}^{L} \sum_{j: i, i'' \in A_j} x_j \leq \sum_{j: i'' \in A_j} x_j \leq 1, \forall i'' \in [N] \backslash \{1, \dots, L\},$$

$$\sum_{i'' \in [N] \backslash [L]} \alpha_{ii''} = \sum_{i'' \in [N] \backslash [L]} \sum_{j: i, i'' \in A_j} x_j = \sum_{j: i \in A_j} \sum_{i'' \in [N] \backslash [L]} x_j \mathbb{1}\{i'' \in A_j\} \leq (L-1) \sum_{j: i \in A_j} x_j \leq (L-1),$$

$$\sum_{i'' \in [N] \backslash [L]} \beta_{ii''} = \sum_{i'' \in [N] \backslash [L]} \sum_{i' \neq i} \sum_{j: i', i'' \in A_j} x_j \leq \sum_{i' \neq i} (L-1) \sum_{j: i' \in A_j} x_j \leq (L-1)^2.$$

Therefore, the optimization problem below provides an upper bound to term (a):

$$\max_{K, \alpha_k, \beta_k} \sum_{k=1}^{K} \alpha_k \beta_k, \text{ s.t. } \alpha_k + \beta_k \leq 1, \forall k, \sum_{k=1}^{K} \alpha_k \leq L-1, \sum_{k=1}^{K} \beta_k \leq (L-1)^2. \tag{20}$$

We claim the optimal value to Problem (20) is $(L-1)^2/L$, which is achieved at $\alpha_k = 1/L$, $\beta_k = 1 - 1/L$ for all k and $K = L(L-1)$. We first show it is sufficient to consider $K^* = L(L-1)$. Suppose $K > L(L-1)$, let $(\alpha_k^*, \beta_k^*)_k$ denote an optimal solution. Without loss of generality, assume $\alpha_1^* \geq \alpha_2^* \geq \dots \alpha_K^*$, then $(\beta_k^*)_k$ is optimal when β_k^* is set as large as possible following the index order until the sum reaches $(L-1)^2$. That is, there exists an index k^* where k^* is the smallest number such that for any $k \geq k^* + 1$, $\beta_k^* = 0$ and

$$\beta_1^* = 1 - \alpha_1^*, \ \beta_2^* = \min\left\{1 - \alpha_2^*, (L-1)^2 - \beta_1^*\right\}, \dots, \beta_{k^*}^* = \min\left\{1 - \alpha_{k^*}^*, (L-1)^2 - \sum_{k=1}^{k^*-1} \beta_k^*\right\}.$$

By the definition of k^*, it holds that

$$\beta_k^* = 1 - \alpha_k^*, \forall k < k^*, \ (L-1)^2 - \sum_{i=1}^{k^*-1} \beta_i^* \leq 1 - \alpha_{k^*}^*,$$

otherwise, if $\beta_k^* \neq 1 - \alpha_k^*$ for some $k < k^*$, then $\beta_{k+1}^* = 0$, contradicting to the fact that k^* is the smallest index. Now suppose $k^* \geq L(L-1)+1$ and $\alpha_{L(L-1)+1}^* > 0$, then we have

$$\beta_{L(L-1)}^* = 1 - \alpha_{L(L-1)}^* \leq (L-1)^2 - \sum_{k=1}^{L(L-1)-1} \beta_k^* = (L-1)^2 - \sum_{k=1}^{L(L-1)-1} (1 - \alpha_k^*),$$

which implies that

$$(L-1)^2 \geq \sum_{k=1}^{L(L-1)} (1 - \alpha_k^*) = L(L-1) - \sum_{k=1}^{L(L-1)} \alpha_k^* > L(L-1) - (L-1) = (L-1)^2,$$

which leads to a contradiction. Thus, we either have $k^* \leq L(L-1)$ or $\alpha_{L(L-1)+1}^* = 0$. In both cases, since $\alpha_k^* \beta_k^* = 0$ for any $k > L(L-1)$, it is sufficient to consider $K^* = L(L-1)$. Therefore, the optimization problem (20) can be reduced to

$$\max_{\alpha_k, \beta_k} \sum_{k=1}^{L(L-1)} \alpha_k \beta_k, \text{ s.t. } \alpha_k + \beta_k \leq 1, \forall k, \sum_{k=1}^{L(L-1)} \alpha_k \leq L-1, \sum_{k=1}^{L(L-1)} \beta_k \leq (L-1)^2.$$

Note that it is sufficient to consider the case where all constraints are tight. Therefore, the problem is equivalent to

$$\min_{\alpha_k} \sum_{k=1}^{L(L-1)} \alpha_k^2, \text{ s.t. } \sum_{k=1}^{L(L-1)} \alpha_k = L-1.$$

Since the problem is to minimize a convex function, we have $\alpha_k^* = 1/L$ for any k. Thus, we can conclude that term (a) is upper bounded by $(L-1)^2$. $\square$

Proof of Lemma 4.4. Analogous to the proof of Lemma 4.2, we have

$$\begin{aligned}
&\sum_{i=1}^{L} \sum_{i' \neq i} \sum_{t=1}^{T} \sum_{t' \neq t} \sum_{\substack{j \in N_t, j' \in N_{t'} \\ A_j \cap A_{j'} = \emptyset \\ i \in A_j, i' \in A_{j'}}} x_j x_{j'} \\
=& \sum_{i=1}^{L} \sum_{i' \neq i} \sum_{t=1}^{T} \sum_{t'=1}^{T} \sum_{\substack{j \in N_t, j' \in N_{t'} \\ A_j \cap A_{j'} = \emptyset \\ i \in A_j, i' \in A_{j'}}} x_j x_{j'} - \sum_{i=1}^{L} \sum_{i' \neq i} \sum_{t=1}^{T} \sum_{\substack{j \in N_t, j' \in N_t \\ A_j \cap A_{j'} = \emptyset \\ i \in A_j, i' \in A_{j'}}} x_j x_{j'}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{\substack{(j,j'):\\ A_j\cap A_{j'}=\emptyset\\ i\in A_j, i'\in A_{j'}}} \left(\left(\sum_{t=1}^{T} x_j \mathbb{1}\{j\in N_t\}\right) \left(\sum_{t=1}^{T} x_{j'} \mathbb{1}\{j'\in N_t\}\right) - \sum_{t=1}^{T} x_j x_{j'} \mathbb{1}\{j,j'\in N_t\} \right) \\
&\overset{\text{(i)}}{=} \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{\substack{(j,j'):\\ A_j\cap A_{j'}=\emptyset\\ i\in A_j, i'\in A_{j'}}} \left(x_j x_{j'} - \sum_{t=1}^{T} x_j x_{j'} \mathbb{1}\{j,j'\in N_t\} \right) \\
&= \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{j:i\in A_j} \sum_{j':i'\in A_{j'}} \left(x_j x_{j'} - \sum_{t=1}^{T} x_j x_{j'} \mathbb{1}\{j,j'\in N_t\} \right) \left(1 - \mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\}\right) \\
&\geq \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{j:i\in A_j} \sum_{j':i'\in A_{j'}} \left(x_j x_{j'} - \sum_{t=1}^{T} x_j x_{j'} \mathbb{1}\{j,j'\in N_t\} - x_j x_{j'} \mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\} \right) \\
&= \sum_{i=1}^{L} \sum_{i'\neq i} \left(\sum_{j:i\in A_j} x_j \right) \left(\sum_{j':i'\in A_{j'}} x_{j'} \right) - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{t=1}^{T} \left(\sum_{j\in N_t:i\in A_j} x_j \right) \left(\sum_{j'\in N_t:i'\in A_{j'}} x_{j'} \right) \\
&\quad - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{j:i\in A_j} \sum_{j':i'\in A_{j'}} x_j x_{j'} \mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\} \\
&= L(L-1) - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{t=1}^{T} \left(\sum_{j\in N_t:i\in A_j} x_j \right) \left(\sum_{j'\in N_t:i'\in A_{j'}} x_{j'} \right) - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{j:i\in A_j} \sum_{j':i'\in A_{j'}} x_j x_{j'} \mathbb{1}\left\{A_j\cap A_{j'}\neq\emptyset\right\} \\
&\geq L(L-1) - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{t=1}^{T} \left(\sum_{j\in N_t:i\in A_j} x_j \right) \left(\sum_{j'\in N_t:i'\in A_{j'}} x_{j'} \right) - \sum_{i=1}^{L} \sum_{i'\neq i} \sum_{j:i\in A_j} \sum_{i''\in A_j\setminus\{i\}} \sum_{j':i',i''\in A_{j'}} x_j x_{j'} \\
&\overset{\text{(ii)}}{=} L(L-1) - \underbrace{\sum_{i=1}^{L} \sum_{i'\neq i} \sum_{t=1}^{T} \left(\sum_{j\in N_t:i\in A_j} x_j \right) \left(\sum_{j'\in N_t:i'\in A_{j'}} x_{j'} \right)}_{(a)} \\
&\quad - \underbrace{\sum_{i=1}^{L} \sum_{i''\in[N]\setminus[L]\cup M_i} \left(\sum_{j:i,i''\in A_j} x_j \right) \left(\sum_{\substack{i'\neq i:\\ i'\notin M_\ell \text{ if } i''\in M_\ell}} \sum_{j':i',i''\in A_{j'}} x_{j'} \right)}_{(b)},
\end{aligned}
$$

where equality (i) holds because N_t are disjoint across time t and equality (ii) holds because all products intersect with items $\{1,\ldots,L\}$ at most once and every product has exactly one item from the set M_ℓ. We now analyze the two terms (a) and (b) separately.

For term (a), let $y_{ti} = \sum_{j\in N_t: i\in A_j} x_j$ for simplicity, then for any $i \in \{1, \dots, L\}$, it holds that

$$\sum_{t=1}^{T} y_{ti} = \sum_{t=1}^{T} \sum_{j\in N_t:i\in A_j} x_j = \sum_{j:i\in A_j} x_j = 1,$$

$$\sum_{t=1}^{T} \sum_{i'\neq i} y_{ti'} = \sum_{t=1}^{T} \sum_{i'\neq i} \sum_{j\in N_t:i'\in A_j} x_j = \sum_{i'\neq i} \sum_{j:i'\in A_j} x_j = L-1,$$

$$\sum_{i=1}^{L} y_{ti} = \sum_{i=1}^{L} \sum_{j\in N_t:i\in A_j} x_j \leq \sum_{j\in N_t} x_j \leq 1, \forall t.$$

To upper bound term (a), for any fixed i, we consider the optimization problem:

$$\max \sum_{t=1}^{T} y_{ti} \left(\sum_{i'\neq i} y_{ti'} \right), \text{ s.t.} \sum_{t=1}^{T} y_{ti} \leq 1, \sum_{t=1}^{T} \sum_{i'\neq i} y_{ti'} \leq (L-1), \sum_{i=1}^{L} y_{ti} \leq 1, \forall t.$$

Similar to the proof of Lemma 4.2, it is sufficient to consider $T = L$, the optimal value is $(L-1)/L$ achieved at $y_{ti} = 1/L$. Thus, it follows that term (a) is upper bounded by $L-1$.

For term (b), note that for any fixed i, since every product j has exactly one item from the set M_ℓ, it holds that

$$\sum_{i''\in[N]\setminus[L]\cup M_i} \sum_{j:i,i''\in A_j} x_j = \sum_{\ell\neq i} \sum_{i''\in M_\ell} \sum_{j:i,i''\in A_j} x_j = \sum_{\ell\neq i} \sum_{j:i\in A_j} x_j = L-1,$$

and

$$\begin{aligned} &\sum_{i''\in[N]\setminus[L]\cup M_i} \sum_{\substack{i'\neq i: \\ i'\notin M_\ell \text{ if } i''\in M_\ell}} \sum_{j':i',i''\in A_{j'}} x_{j'} \\ =& \sum_{\ell\neq i} \sum_{i''\in M_\ell} \sum_{i'\notin\{i,\ell\}} \sum_{j':i',i''\in A_{j'}} x_{j'} = \sum_{\ell\neq i} \sum_{i'\notin\{i,\ell\}} \sum_{j':i'\in A_{j'}} x_{j'} = (L-2)(L-1). \end{aligned}$$

Moreover, for any fixed i'', we have

$$\sum_{j:i,i''\in A_j} x_j + \sum_{\substack{i'\neq i: \\ i'\notin M_\ell \text{ if } i''\in M_\ell}} \sum_{j':i',i''\in A_{j'}} x_{j'} = \sum_{j:i''\in A_j} x_j \leq 1.$$

For simplicity, let

$$\alpha_{ii''} = \sum_{j:i,i'' \in A_j} x_j, \ \beta_{ii''} = \sum_{\substack{i' \neq i: \\ i' \notin M_\ell \text{ if } i'' \in M_\ell}} \sum_{j':i',i'' \in A_{j'}} x_{j'},$$

for any fixed i, we consider the optimization problem:

$$\begin{aligned} \max & \sum_{i'' \in [N] \setminus [L]} \alpha_{ii''} \beta_{ii''}, \\ \text{s.t.} & \sum_{i'' \in [N] \setminus [L]} \alpha_{ii''} = L-1, \forall i, \sum_{i'' \in [N] \setminus [L]} \beta_{ii''} = (L-2)(L-1), \forall i, \alpha_{ii''} + \beta_{ii''} \leq 1, \forall i, i''. \end{aligned}$$

Again, similar to the proof of Lemma 4.2, the optimal value is $L-2$ which is obtained when $\alpha_{ii''} = 1/(L-1)$ and $\beta_{ii''} = (L-2)/(L-1)$, and it then follows that term (b) is upper bounded by $L(L-2)$.

In conclusion, we have that

$$\sum_{i=1}^{L} \sum_{i' \neq i} \sum_{t=1}^{T} \sum_{t' \neq t} \sum_{\substack{j \in N_t, j' \in N_{t'} \\ A_j \cap A_{j'} = \emptyset \\ i \in A_j, i' \in A_{j'}}} x_j x_{j'} \geq L(L-1) - (L-1) - L(L-2) = 1.$$

□

C Supplement to Section 5

C.1 Second Initial Processing Step

Here we describe how to transform an abstract problem with substitutable actions into a problem fitting into the OCRS framework. Let $(x_t(S))_{t,S}$ denote an optimal solution to the LP relaxation (5.4). To start with, we first label the initial products $j = 1, \ldots, N$ and items $i = 1, \ldots, M$, and relabel each unit of items, e.g., let i_k denote the k-th unit for item i. Throughout this section, we treat different units of the same item as "different" items so that all items have an initial inventory of 1. Algorithm 1 describes the processing step in detail. Put it briefly, we split original items with multiple initial inventories into items with initial inventory 1 and then we reallocate all active probability $x_j = \sum_{S \in \mathcal{S}_t} \phi_t(j, S) x_t(S)$ into items by creating dummy products (j_ℓ denotes

ℓ-th product j) if necessary.

Algorithm 1: Second Initial Processing Step

Input : $N_t = \emptyset, \forall t$, $\ell(j) = 1, \forall j$, $k(i) = 1, \forall i$, $c_{i_k} = 1, \forall i_k$.

1 For $t = 1, \ldots, T$

2   For $j = 1, \ldots, N$

3   Let $x_j = \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S)$

4   While $x_j > 0$

5   If $\min_{i \in A_j} c_{i_{k(i)}} \geq \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S)$

6                 $c_{i_{k(i)}} \leftarrow c_{i_{k(i)}} - \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S)$, $\forall i \in A_j$

7                 $x_j \leftarrow 0$, $A_{j_{\ell(j)}} = \cup_{i \in A_j} \{k(i)\}$ and $N_t \leftarrow N_t \cup \{j_{\ell(i)}\}$

8   Else

9                 $\delta = \min_{i \in A_j} c_{i_{k(i)}} - \sum_{S \in S_t} \phi_t(j,S) x_t(S)$

10                 $c_{i_{k(i)}} \leftarrow c_{i_{k(i)}} - \delta$, If $c_{i_{k(i)}} = 0$, $k(i) \leftarrow k(i) + 1, \forall i \in A_j$

11                 $x_j \leftarrow x_j - \delta$, $\ell(j) \leftarrow \ell(j) + 1$, $N_t \leftarrow N_t \cup \{j_{\ell(j)}\}$

12   End

13   End

14   End (j)

15 End (t)

Output: $N_t, \forall t, A_{j_\ell}, \forall j_\ell$.

By Algorithm 1, it follows immediately that the active probabilities of all products satisfy the feasibility constraints in expectation and the sum of active probabilities per period is less than 1. Therefore, the fluid relaxation of the reduced problem provides an upper bound to the original problem, and for any policy provides a constant approximation to this problem against the corresponding fluid LP, it provides same constant approximation to the original problem.

Furthermore, note that a dummy product is created only if an unit is "overflowed" (Step 10 and 11 in Algorithm 1). In addition, it holds that $\sum_{S \in \mathcal{S}_t} x_t(S) = 1, \forall t$. Therefore, for every period t, it holds that

$$\sum_{j: i \in A_j} \sum_{S \in \mathcal{S}_t} \phi_t(j,S) x_t(S) = \sum_{S \in \mathcal{S}_t} \left(\sum_{j: i \in A_j} \phi_t(j,S) \right) x_t(S) \leq \sum_{S \in \mathcal{S}_t} x_t(S) = 1, \forall i \in M,$$

which implies that each item $i \in M$ can be consumed for at most one unit and thus there can be at most one "overflow" for each item i. Hence, there are at most M dummy products created for each period, which implies that the reduced problem is still polynomial sized.

Finally, if the original problem instance had no substitution, then by definition, for any time step t the products j that can have $x_j > 0$ must all have identical item sets A_j. Therefore, any j

that Algorithm 1 can add to N_t consumes the same bundle of items, allowing us to apply standard OCRS.

C.2 Proof of Theorem 5.1

Recall that $x_j = \sum_{S\in\mathcal{S}_t} \phi_t(j,S)x_t(S)$ for all $t = 1,\dots,T$ and $j \in N_t$. By the assumptions of the transformed instance, LP constraint (13) implies that $\sum_{t=1}^{T}\sum_{j\in N_t:i\in A_j} x_j \le 1$ for all $i \in M$, and LP objective (12) equals $\sum_{t=1}^{T}\sum_{j\in N_t} r_j x_j$. For all t, we also have that

$$\begin{aligned}\sum_{j\in N_t} x_j &= \sum_{S\in\mathcal{S}_t} x_t(S) \sum_{j\in N_t} \phi_t(j,S)\\ &\le \sum_{S\in\mathcal{S}_t} x_t(S)\cdot(1)\\ &= 1,\end{aligned}$$

where the inequality applies the assumption that $\sum_j \phi_t(j,S) \le 1$ in Definition 5.1, and the final equality applies LP constraint (14).

Therefore, vector $(x_j)_{j\in N}$ satisfies the conditions of a random-element OCRS for L-bounded products. The OCRS, if α-selectable, is able to accept every j w.p. αx_j, while only accepting active products and satisfying the item feasibility constraints. This can be re-interpreted as follows. For each t and $j \in N_t$, let X_j indicate whether j is active, i.e. $\mathbb{E}[X_j] = x_j$ and $\sum_{j\in N_t} X_j \le 1$ w.p. 1. For each t, based on its present state, the OCRS can *pre-decide* whether to accept each product $j \in N_t$ if it were to be active, indicated by $B_j \in \{0,1\}$. Product j is then accepted if and only if $B_jX_j = 1$. The OCRS guarantees that $\mathbb{E}[B_jX_j] = \alpha x_j$, which equals $\mathbb{E}[B_j]\mathbb{E}[X_j]$ because X_j is independent from everything else. Cancelling because $\mathbb{E}[X_j] = x_j$, we deduce that $\mathbb{E}[B_j] = \alpha$.

We use these random bits $(B_j)_{j\in N_t}$ in the online algorithm. As indicated in step 2 of the algorithm, conditional on $(B_j)_{j\in N_t}$, we would like to play a randomized action so that the probability of selling each product $j \in N_t$ is x_jB_j. To show that this is possible, recall that $x_j = \sum_{S\in\mathcal{S}_t} \phi_t(j,S)x_t(S)$. For each $S \in \mathcal{S}_t$, we apply Lemma 5.2 with forbidden product set $F := \{j \in N_t : B_j = 0\}$ to find a randomized recourse action S' such that $\mathbb{E}_{S'}[\phi_t(j,S')] = \phi_t(j,S)B_j$. Therefore, if we play the mixture of randomized resource actions S' for different original actions $S \in \mathcal{S}_t$

weighted by $x_t(S)$, then the probability of selling each product $j \in N_t$ would be the desired

$$\sum_{S \in \mathcal{S}_t} x_t(S)(\phi_t(j,S)B_j) = x_j B_j.$$

Finally, we must show that the state evolution in the actual problem is consistent with the state evolution expected by the OCRS. We can define the following coupling: in the actual problem, for each t, product $j \in N_t$ is sold if and only if $X_j B_j = 1$. Conditional on any realization $(B_j)_{j \in N_t}$, we will indeed see that product j is sold w.p. 0 if $B_j = 0$, and w.p. x_j if $B_j = 1$, correlated across j so that at most one product is sold. Moreover, the realization of which product (if any) is sold is independent from everything else, which is consistent with the desired state evolution in the actual problem. Therefore, the state in the actual problem (where we cannot see whether products are "active" before deciding accept/reject) can be coupled with the state in the OCRS, and hence the OCRS guarantee which implies $\mathbb{E}[B_j] = \alpha$ for all j can be applied. Moreover, the OCRS guarantees that $B_j = 0$ whenever j is infeasible, leading to a valid algorithm in the actual problem that respects the inventory constraints. This completes the proof.

C.3 Proof of Lemma 5.2

For any action $S \in \mathcal{S}_t$ and set of forbidden products $F \subseteq N_t$, since ϕ_t defines substitutable actions, there exists an action $S_1 \in \mathcal{S}_t$ such that

$$\phi_t(j, S_1) = 0, \forall j \in F, \text{and } \phi_t(j, S_1) \geq \phi_t(j, S), \forall j \notin F.$$

Let $J_1 = \operatorname{argmin}_{j \notin F} \phi_t(j,S)/\phi_t(j,S_1)$, $\gamma_1 = \min_{j \notin F} \phi_t(j,S)/\phi_t(j,S_1)$ and $F_1 = F \cup J_1$. Note that if there does not exist a product $j \notin F$ such that $\phi_t(j,S_1) > \phi_t(j,S)$, then by definition, we have $\gamma_1 = 1$, $J_1 = N_t \backslash F$ and the action S_1 satisfies the conditions. Suppose not, then $N_t \backslash F_1 \neq \emptyset$ and we proceed to the next iteration. Now consider the action S_1 and the set F_1, again by the substitutable assumption, there exists an action S_2 such that

$$\phi_t(j, S_2) = 0, \forall j \in F_1, \text{ and } \phi_t(j, S_2) \geq \phi_t(j, S_1) \geq \phi_t(j, S), \forall j \notin F_1.$$

Similarly, let

$$J_2 = \operatorname*{argmin}_{j \notin F_1} \left(\phi_t(j, S) - \gamma_1 \phi_t(j, S_1) \right) / \phi_t(j, S_2),$$

$$\gamma_2 = \min_{j \notin F_1} \left(\phi_t(j, S) - \gamma_1 \phi_t(j, S_1) \right) / \phi_t(j, S_2),$$

and $J_2 = F_1 \cup J_1$. This process is repeated until the end of K-iteration if $F_K = N_t$. Note that the set $N_t \backslash F$ is finite and we remove at least one element in each iteration, therefore, this process must terminate within finite steps.

Suppose the process terminates at K-th iteration. We now consider the randomized action S' which offers action S_k with probability γ_k. We claim it is a well-defined randomized action which satisfies the conditions we want. In order to show it is a well-defined randomized action, we need to show $\gamma_k \geq 0$ for any $k \in [K]$ and $\sum_{k=1}^{K} \gamma_k \leq 1$. We show the result by induction. Note that it holds that $0 \leq \gamma_1 \leq 1$. Suppose that $\gamma_{k'} \geq 0$ for any $k' \leq k$ and $\sum_{k'=1}^{k} \gamma_{k'} \leq 1$, now for $k+1$-th iteration, we have

$$\gamma_{k+1} = \min_{j \notin F_k} \frac{\phi_t(j, S) - \sum_{k' \leq k} \gamma_{k'} \phi_t(j, S_{k'})}{\phi_t(j, S_{k+1})} = \frac{\phi_t(j_k, S) - \sum_{k' \leq k} \gamma_{k'} \phi_t(j_k, S_{k'})}{\phi_t(j_k, S_{k+1})},$$

where $j_k \in J_{k+1}$. Note that $J_{k+1} \subseteq N_t \backslash F_k$, thus $j_k \notin F_{k-1}$ and

$$\gamma_k = \min_{j \notin F_{k-1}} \frac{\phi_t(j, S) - \sum_{k' \leq k-1} \gamma_{k'} \phi_t(j, S_{k'})}{\phi_t(j, S_k)} \leq \frac{\phi_t(j_k, S) - \sum_{k' \leq k-1} \gamma_{k'} \phi_t(j_k, S_{k'})}{\phi_t(j_k, S_k)},$$

thus, it follows that

$$\phi_t(j_k, S) - \sum_{k' \leq k} \gamma_{k'} \phi_t(j_k, S_{k'}) \geq 0,$$

and $\gamma_{k+1} \geq 0$. By our construction, it holds that for any product $j \notin F_k$,

$$\phi_t(j, S_{k+1}) \geq \phi_t(j, S_k) \geq \cdots \geq \phi_t(j, S_1) \geq \phi_t(j, S),$$

therefore, it holds that

$$\sum_{k' \leq k} \gamma_{k'} \phi_t(j, S_{k+1}) - \sum_{k' \leq k} \gamma_{k'} \phi_t(j, S_{k'}) \leq \sum_{k' \leq k} \gamma_{k'} \left(\phi_t(j, S_{k+1}) - \phi_t(j, S) \right) \leq \phi_t(j, S_{k+1}) - \phi_t(j, S),$$

which implies

$$\phi_t(j,S) - \sum_{k' \le k} \gamma_{k'} \phi_t(j, S_{k'}) \le \phi_t(j, S_{k+1}) \left(1 - \sum_{k' \le k} \gamma_{k'}\right),$$

and thus

$$\gamma_{k+1} \le 1 - \sum_{k' \le k} \gamma_{k'}.$$

Hence, we can conclude that the randomized action S' is indeed well-defined. Finally, by our construction, it holds that $\{J_k\}_k$ forms a partition to the set $N_t \backslash F$. For any product $j \in F$, since $\phi_t(j, S_k) = 0$ for any k, thus $\mathbb{E}_{S'}[\phi_t(j, S')] = 0$. For any product $j \notin F$, there exists k-th iteration so that $j \in J_k$ and by definition,

$$\gamma_k = \frac{\phi_t(j,S) - \sum_{k' \le k-1} \gamma_{k'} \phi_t(j, S_{k'})}{\phi_t(j, S_k)},$$

and thus $\mathbb{E}_{S'}[\phi_t(j, S')] = \phi_t(j, S)$ because $\phi_t(j, S_{k'}) = 0$ for any $k' \ge k+1$. This completes the proof.