Appendix A: Examples of Demand Data Sets

In this section, we conduct simple analyses to understand the extent to which the assumption of serial independence (SI), standard in online stochastic matching, is valid on real demand data. We consider two distinct data sets, a publicly available data set from the online retailer JD.com (Shen et al. 2020) and a private data set from a fashion e-commerce platform. In both settings, the platform needs to fulfil customer orders for various products by dispatching them from different warehouses to different locations. Because formally testing the serial independence (SI) assumption is not straightforward, we report various model-free and regression-based analyses. Recall that a consequence of the serial independence assumption is that $\mathrm{Var}(D_j) \leq \mathbb{E}[D_j]$ for the demands of each type j. In the e-commerce context, each j corresponds to a certain type of customer request based on the selected product and the closest warehouse to the customer location. We find that $\mathrm{Var}(D_j) \leq \mathbb{E}[D_j]$ does not hold for a majority of types when comparing the sample mean and sample variance of the realized demand aggregated over a plausible replenishment horizon. In addition, we test for correlations in the panel data set formed by daily demand per type. We estimate fixed effects regressions and still observe that the sample variance of residual errors over query types often exceeds the corresponding sample mean demand.

A.1. JD.com data

Context and data description. The online matching problem faced by JD.com consists of dynamically dispatching customer orders containing a stock-keeping unit (SKU) from various locations to different fulfillment centers (that still have inventory of that SKU) over time. Higher rewards are obtained when orders are dispatched to nearby distribution centers, resulting in an edge-weighted online matching problem.

The JD.com data provides customer orders throughout China from a single product category in March 2018. We construct a dataset with daily demand level for each (SKU-location) combination. We focus our analysis on the 40 highest-selling SKUs and 40 largest locations, determined by the destination fulfillment center in China. When choosing the 40 highest-selling SKUs, we eliminate any SKU that had zero demand on any day because they may suggest an inventory stockout. We otherwise assume that enough inventory was available so that the observed sales coincides with the true demand. Second, we remove the first day of sales, which we found to be three times higher than an average day due to promotions. After this processing, we let D_j^t denote the demand for a type j on a day t, where a type j refers to a (SKU-location) combination.

Model-free evidence. We compare the sample mean and sample variance for the demand random variable D_j across types. In stochastic matching models, the time horizon represents the duration between consecutive inventory replenishments. Thus, since our analysis focuses on highest-selling items, we consider this duration to be equal to be one week in JD.com context: a realization of the demand D_j is the total amount that customers ordered in a week, with the type j referring to a particular SKU and a particular location. Consequently, we aggregate our demand data set D_j^t at the weekly level. Each week w yields a sample of D_j^w for every (SKU, location)-combination j. We evaluate the sample mean and sample variance of D_j^w for each type j. In the resultant scatter-plot Figure 5, we visualize the log-ratio of sample variance to sample mean (y-axis) as a function of the sample mean (x-axis).

The property $\mathrm{Var}(D_j) \leq \mathbb{E}[D_j]$ would imply that the dots in the scatter-plot Figure 5 are below the $y = 0$ line with high probability. By contrast, we find that this is not the case and sample variance is greater than sample mean for a majority of types (85%), often by orders of magnitude. Only for a minority of (SKU, location)-combinations is the empirical variance lower than the empirical mean. This suggests that standard online stochastic matching models might not accurately reflect the stochastic demand faced by JD.com. Counter-intuitively, the property $\mathrm{Var}(D_j) \leq \mathbb{E}[D_j]$ appears to hold for SKUs j with small mean demand rather than large mean demand. Considering the fact that the sample mean and sample variance are estimated from only 1 month of weekly sales data and therefore are noisy, we simulate this experiment on synthetic data where the demand is Poisson-distributed, and thus, $\mathrm{Var}(D_j) \leq \mathbb{E}[D_j]$. We find that the probability that sample variance exceeds sample mean is of approximately 39%. Hence, the noise in our estimates cannot explain the fact a majority of types have sample variance above the $y = 0$ line.

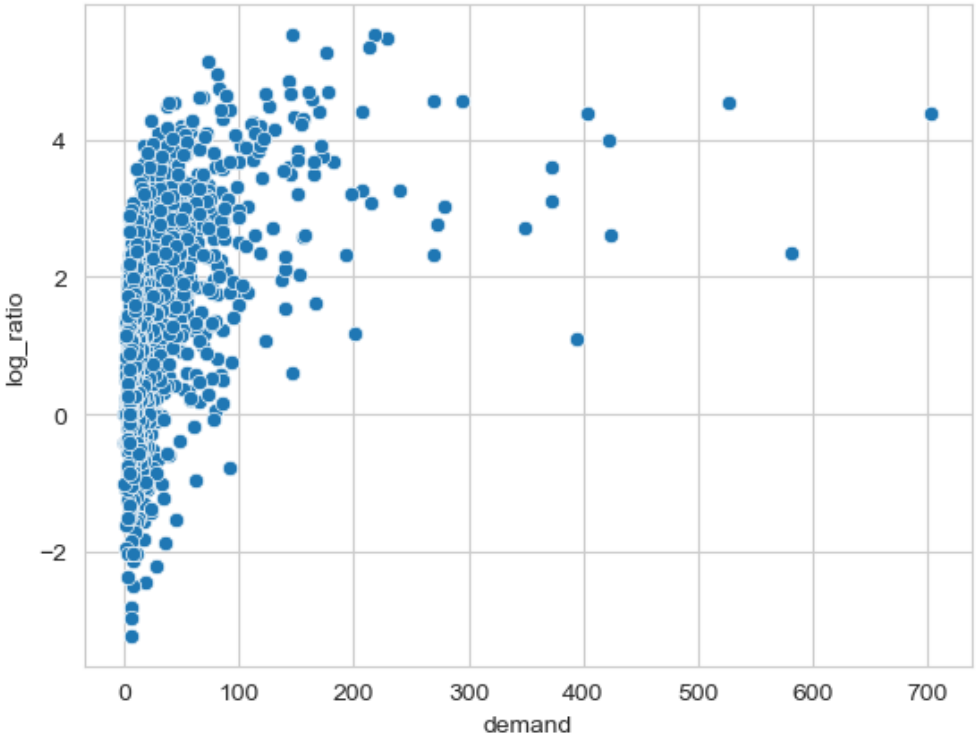


Figure 5 Log-ratio mean-variance plots for the JD.com data set. Each dot represents a (SKU, location)-type j. The x-coordinate is the sample mean demand for that SKU from that location, and the y-coordinate is the natural logarithm of the ratio of sample variance to the sample mean.

Our analysis suggests that demand data violates the basic SI assumption. That said, the variance of SKU-level demands can be explained by several factors, including ones that can be anticipated by the retailer in their forecasts. It is plausible that after controlling for contextual factors like marketing promotions and calendar events, the "unexplained" variance in demand might be lower than what is inferred in Figure 5. Another important limitation of our analysis is that we only have one month of sales data and one product

category, which is a restriction of the JD.com data. Thus, we conduct a similar analysis in the next subsection using a data set with a longer time window and more detailed information.

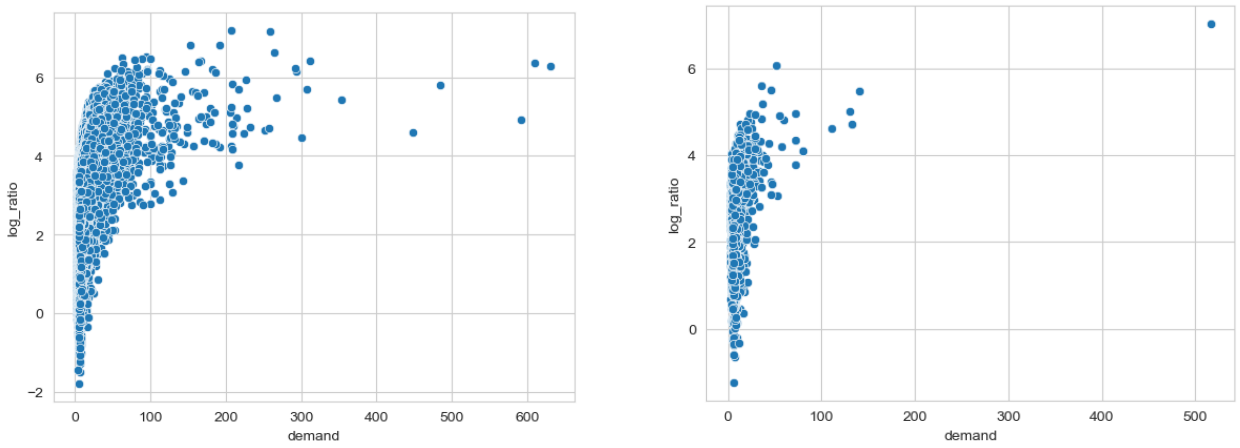
A.2. Fashion retail data

Context and data description. We replicate the analysis of Appendix A.1 on a different data set, describing customer order fulfillment for a major fashion e-commerce platform in India. The online matching problem is similar to that of JD.com but the data and context are very different. While our analysis of Appendix A.1 was restricted to one product category over 1 month, the raw data describes customer orders from 11,800 location IDs over 12 months for 1,256,241 distinct SKUs across all categories. Products are dispatched from one of 14 warehouses. We do not know which fulfillment center is the closest to the customer requests. That said, it is plausible that the platform prefers to dispatch from the closest warehouse, unless inventory is low or depleted, in which case it might choose a further warehouse. Similarly to Appendix A.1, we treat warehouse ID as our unit of geographic aggregation. That is, customer requests are classified into demand types, each corresponding to a SKU and the closest warehouse.

Similarly to Appendix A.1, we construct a panel dataset with demand level D_j^t for each (SKU-location) combination (or type) j on each day t. We drop all orders that have been subsequently cancelled, after which we focus our analysis on the 100,000 highest-selling types. SKUs in fashion retail may have short product cycles, hence we truncate the dataset to only consider any given SKU from the moment it has recorded at least one transaction until its last recorded transaction. Finally, although customers might order multiple units of the same SKU, the demand D_j^t is the count of unique customer orders on day t, rather than the total ordered quantity. This assumption is conservative, as large order quantities mechanically inflate the variance-mean demand ratio. Our final data set D_j^t comprises 16,046,034 demand observations for 100,000 distinct types j with 77,938 SKUs and 12 warehouse locations.

Model-free evidence. We compare the sample mean and sample variance of the demand random variable D_j across types. In fashion retail, the replenishment periods are less frequent, even for high-selling SKUs. Hence, we take the time horizon of the online matching problem to be equal to one month rather than 1 week in the case of JD.com data. We aggregate our demand data set D_j^t at that granularity. Each month τ yields a sample of D_j^τ for every (SKU, location)-type j. We plot the log-ratio of sample variance and sample mean of D_j^τ for each type j in Figure 6a. Similarly to our analysis of JD.com data, we find that the sample variance tends to be larger than the sample mean for nearly all types (99.4%).

Controlling for variability. In practice, the e-commerce platform can anticipate and forecast various sources of variability, reducing the uncertainty in the demand. We consider a simple two-way fixed effect model $D_j^\tau = \alpha_j + \beta_\tau + \gamma_j P_{j,\tau} + \varepsilon_{j,\tau}$, where α_j is a type fixed effect, β_τ is the month fixed effect, $P_{j,\tau}$ is a per-unit sales-weighted price for type j demand in month τ, and $\varepsilon_{j,\tau}$ is the residual error. This model controls for variability explained by price variation at the type-level and common seasonal patterns across types. Prices are inferred using as input the time-series of maximum retail price (MRP) for each item—for each SKU, we backfill missing prices chronologically, then normalize price as log-ratio between current price and the median non-zero price for that SKU.



(a) (b)

Figure 6 Log-ratio mean-variance plots for the fashion e-commerce data set. Each dot represents a (SKU, location)-type j. The x-coordinate is the sample mean of the monthly aggregated demand for that SKU from that location. The y-coordinate is the natural logarithm of the ratio of sample variance to the sample mean. In (a), we compute the sample variance of monthly demands per type. In (b), we compute the sample variance of the linear model's residual errors per type (the linear model is estimated on a subsample of 1,500 types). In both cases, we restrict attention to SKUs offered for at least 6 months.

We estimate the variance of (in-sample) residual errors $\varepsilon_{j,\tau}$ for each type j and month τ in our panel. Due to the scale of the data, we conduct our regression on random subsamples of 1,500 types (we repeat our analysis on multiple subsamples and the findings are unchanged). The log-ratio mean-variance plot is shown in Figure 6b. Despite better explaining variability in the demand, our linear model does not significantly reduce the level of uncertainty. The vast majority of (SKU-location) types still violate the SI assumption.

Limitations of our analysis. Our analysis has several limitations. Selecting the top 100,000 (SKU, region) pairs by total demand during the study period focuses our analysis on a specific sample. High-volume types may systematically differ in variance properties from the full population.

The linear model does not capture non-linear effects. Seasonal shifts in demand may be anticipated for subcategories of SKUs (i.e., a refined model may use month-level fixed-effects per subcategory of SKUs instead of β_t-s). Our analysis overlooks other SKU-specific marketing interventions, known to the retailer, that affect the base demand. The MRP may not reflect true transaction price. Moreover, products in fashion retailing have a short product life cycle and strong seasonality, meaning that some of the variation in demand could be due to planned assortment rotations. Unfortunately, this information was not available to us in the dataset (i.e., we do not know when a SKU is offered and when it is discontinued). Note that we pre-processed the data to only consider "active" SKUs at any point in time using a simple heuristic, i.e., each (SKU-location) is included in the panel from the time it records at least one transaction until its last transaction.

Lastly, we do not have access to inventory records, and thus, a zero demand may be due to stock-outs—something we do not control for.

Appendix B: Connection to Standard Models

We call Poisson the online stochastic matching problem in continuous time where each type $j \in [m]$ arrives following an independent Poisson process of rate λ_j over the time horizon $[0,1]$. We now explain why both our Indep and Correl models may capture Poisson as a special case. To reveal the connection with Poisson, suppose we operate under the random order Rand, but we further assume that each arrival has a timestamp in the horizon $[0,1]$.[10] Under the Poisson model, we note that the demand D_j of each type j ends up having a distribution that is Poisson with expectation λ_j. Moreover, a basic property of stationary marked Poisson processes is that the density of arrivals is uniform over $[0,1]$ conditional on each realization of $\mathbf{D}$. This implies that any instance of Poisson with arrival rates $(\lambda_j)_{j\in[m]}$ can be captured within our framework by Indep with random order, assuming that each demand entry D_j follows a Poisson distribution with mean $\mathbb{E}[D_j] = \lambda_j$. Similarly, the splitting property of Poisson processes straightforwardly implies that this Poisson instance has an equivalent representation within Correl, where D is Poisson-distributed with mean $\mathbb{E}[D] = \lambda_1 + \cdots + \lambda_m$ and the type probabilities are $p_j = \lambda_j / \mathbb{E}[D]$ for each $j \in [m]$.

We now explain why our results for Indep and Correl can capture the SI model beyond the limiting Poisson case. In the case of Correl, the reduction is straightforward. For a fixed horizon $T = D$ equal to the total demand, the type of the t-th query is drawn according to a distribution over types $(p_{t,j})_{j\in[m]}$, noting that our approximation ratio result for Correl $\cap$ Rand extends to such time-varying distributions; see Appendix E. In the case of Indep, the reduction is less straightforward but one possible approach may be to first link general SI to Poisson, as has been done by Huang and Shu (2021). However, their link only applies to type distributions that IID over time. For time-varying distributions, the demand D_j of each type $j \in [m]$ follows a Poisson binomial distribution with T trials and a success probabilities $(p_{t,j})_{t\in[T]}$, and the challenge is that the resulting demand vector $\mathbf{D}$ does not satisfy the cross-sectional independence property required by Indep. In fact, it is well known that the random variables D_j are *negatively associated* (Dubhashi and Ranjan 1998), which means that $\mathbb{E}[f(D_j, j \in J) \cdot g(D_j, j \in [m] \setminus J)] \le \mathbb{E}[f(D_j, j \in J)]\mathbb{E}[g(D_j, j \in [m] \setminus J)]$ for every non-decreasing functions f, g and every $J \subseteq [m]$. This strong negative dependence property, however, allows us to easily extend our analysis of the 1/2-competitive algorithm for the Indep model (Algorithm 1 in Section 4). In particular, we remark that the formulation of the truncated LP $\mathsf{LP}^{\mathsf{trunc}}$ is unaffected by the dependence of the demand D_j random variables. Thus, our lossless rounding scheme can be separately applied to each demand type to create a reduction to a prophet inequality setting, which is robust to weak notions of negative dependence for the sequence of random rewards (Samuel-Cahn 1991). For completeness, we provide a proof below. Here, we use the weaker notion of Negative Right Orthant Dependence (NOD) property for a sequence of real-valued random variables $X_1, \ldots, X_m$, which requires that $\Pr[X_j \ge t_j,\ \forall j \in J \mid X_k \le t_k,\ \forall k \in [m] \setminus J] \ge \Pr[X_j \ge t_j, \forall j \in J]$ for every vector $\boldsymbol{t} \in \mathbb{R}^m$ and $J \subseteq [m]$; see Dubhashi and Ranjan (1998, Prop. 3).

[10] This can be achieved by assuming that each of the $D = D_1 + \cdots + D_m$ total queries draws an independent arrival time uniformly from horizon [0,1]. This time-based process is combinatorially equivalent to Rand. However, when time is an observable state variable, the online stochastic matching problem is different because a policy can set finer expectations for future demand based on which absolute time in the horizon $[0,1]$ it has reached.

CLAIM 2. *Suppose that $X_1, \ldots, X_n$ is a sequence of NOD two-outcome random variables with rewards $0 = r_0 < r_1 < r_2 < \ldots < r_n$ and probabilities $\Pr[X_i = r_i] = p_i$ and $\Pr[X_i = 0] = 1 - p_i$ such that $\sum_{i=1}^{n} p_i \leq 1$. The static threshold $\tau^* = \frac{1}{2}\sum_{i=1}^{m} p_i r_i$ yields the prophet inequality $\mathbb{E}[R(\tau^*)] \geq \frac{1}{2}(\sum_{i=1}^{n} r_i p_i)$, where $R(\tau^*)$ is equal to the first outcome above τ^*, if any, and otherwise to zero.*

Proof. Let $\theta_i = \Pr[X_i \geq \tau^* | X_k < \tau^*,\ \forall k \leq i-1]$. By the NOD property, we clearly have that $\theta_i \geq \Pr[X_i \geq \tau^*] = p_i \cdot \mathbb{I}[r_i \geq \tau^*]$. Denoting by $\theta = \Pr[X_i < \tau^*\ , \forall i \in [n]]$, we can decompose the expected reward of the τ^*-static threshold policy as

$$\begin{aligned}
\mathbb{E}[R(\tau^*)] &= \sum_{i=1}^{n} (r_i - \tau^*)\theta_i \Pr\left[X_k < \tau^*,\ \forall k \leq i-1\right] + \sum_{i=1}^{n} \tau^* \theta_i \\
&\geq \sum_{i=1}^{n} (r_i - \tau^*) p_i \cdot \mathbb{I}[r_i \geq \tau^*] \cdot \theta + \tau^*(1-\theta) \\
&\geq \theta \cdot \left(\sum_{i=1}^{n} r_i p_i - \tau^* p_i \right) + \tau^*(1-\theta) \\
&\geq \theta \cdot \frac{1}{2} \left(\sum_{i=1}^{n} r_i p_i \right) + \frac{1}{2}(1-\theta) \left(\sum_{i=1}^{n} r_i p_i \right) \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} r_i p_i \right)
\end{aligned}$$

where in the first inequality we use the fact that $\Pr[X_k < \tau^*,\ \forall k \leq i-1] \geq \theta$ for all $i \in [n]$. □

Appendix C: Approximation Algorithm for the CORREL Model

In this section, we focus on the CORREL model. First, we observe in Appendix C.1 that instances in CORREL ∩ RAND are connected to the online stochastic matching problem with a stochastic horizon length. Appendix C.2 introduces and justifies the "conditional" LP that we use as benchmark. Appendix C.3 defines our algorithm based on the conditional LP and establishes our results.

C.1. Connection to the stochastic horizon setting

As mentioned previously, the CORREL model with random order RAND can be interpreted as a variant of the SI model with stochastic horizon. In the stochastic horizon setting, there is a maximum horizon length of T. We are given a vector $(p_j)_{j \in [m]}$ satisfying $\sum_{j=1}^{m} p_j = 1$ and denoting the probabilities that each arriving query is of type j. As per the SI model, the type of each new query in time $t = 1, \ldots, T$ is drawn independently of the history. However, the true horizon length D is unknown but randomly drawn from a known distribution with maximum value T. This means that the arrivals of queries suddenly "stop" after time $t = D$, i.e., the sequence is truncated at time $t = D$ and subsequent queries $t = D+1, \ldots, T$ never occur.

This stochastic horizon setting coincides with CORREL ∩ RAND. To see why, recall that in CORREL ∩ RAND, the total demand D is first drawn from a known distribution, after which D types are drawn IID from probability vector $(p_j)_{j \in [m]}$ and arrive in a random order. However, the random permutation yields a sequence of the same distribution, because the D draws are ex-ante identical. As such, it is equivalent to number the draws $t = 1, \ldots, D$ and assume they arrive in that order, coinciding exactly with the aforementioned setting with the stochastic horizon D.

This stochastic horizon setting has been previously studied by Alijani et al. (2020), in the context of a single item that perishes after an unknown random time D. They show that the reward of the best online algorithm can be an arbitrary factor worse than that of a benchmark that knows the realization of D in advance; i.e.,

$$\inf_{\mathcal{I}\in\textsc{Correl}\cap\textsc{Rand}} \frac{\mathsf{OPT}(\mathcal{I})}{\mathsf{OFF}(\mathcal{I})} = 0 \ .$$

This result has also been discovered by Brubach et al. (2023), Balseiro et al. (2023) in different contexts. Because $\mathsf{LP}(\mathcal{I}) \geq \mathsf{LP}^{\mathsf{trunc}}(\mathcal{I}) \geq \mathsf{OFF}(\mathcal{I})$ for all instances $\mathcal{I}$, neither the commonly-used fluid relaxation nor the LP from Section 4 can provide a meaningful benchmark to analyze the performance of online algorithms. We propose a second, tightened LP benchmark that allows for a constant-factor approximation ratio, comparing to the optimal online algorithm's reward $\mathsf{OPT}(\mathcal{I})$.

C.2. Formulation of the conditional LP

Definition 2. For any instance $\mathcal{I}$, we define $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ as the optimal objective value of the following "conditional" LP:

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{t=1}^{T} \Pr[D \geq t] \cdot r_{i,j} y_{i,j}^{t} \tag{15}$$

$$\text{s.t.} \quad \sum_{j=1}^{m}\sum_{t=1}^{T} y_{i,j}^{t} \leq k_i \qquad \forall i \in [n] \tag{16}$$

$$\sum_{i=1}^{n} y_{i,j}^{t} \leq p_j \qquad \forall j \in [m], \forall t \in [T] \tag{17}$$

$$y_{i,j}^{t} \geq 0 \qquad \forall i \in [n], \forall j \in [m], \forall t \in [T] \ . \tag{18}$$

Similar to Section C.1, the parameter T in $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ stands for the maximum value of D that occurs with non-zero probability. Each decision variable $y_{i,j}^{t}$ represents the probability of matching a query of type j to resource i at time t, *conditional* on $D \geq t$. This explains the objective function (15), as well as constraint (17), in which p_j is the probability that a type j arrival occurs for time t conditional on $D \geq t$. Finally, constraint (16), although appearing at first sight to be missing a coefficient $\Pr[D \geq t]$, is justified by the fact that each resource i can be matched at most k_i times conditional on $D = T$—the maximum value D can ever take. Because any online algorithm cannot foretell the realization of D, conditioning on $D = T$ should be equivalent to conditioning on $D \geq t$ for any time $t \in [T]$. This informally justifies why $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ is a valid benchmark; the claim is formally proved in Appendix C.5.

Lemma 3. *For any instance $\mathcal{I} \in \textsc{Correl}\cap\textsc{Rand}$ and any online algorithm, we have $\mathsf{ALG}(\mathcal{I}) \leq \mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$. Therefore, $\mathsf{OPT}(\mathcal{I}) \leq \mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$.*

It is instructive to compare $\mathsf{LP}^{\mathsf{cond}}$ to the fluid LP in the same setting. The fluid LP has an additional factor of $\Pr[D \geq t]$ on the left-hand side of (16), i.e., $\sum_{i=1}^{n} \Pr[D \geq t] \cdot y_{i,j}^{t} \leq p_j$. This is a relaxation of constraint (16), saying that the *expected consumption* of each resource i cannot exceed k_i—a condition that holds for the offline optimum. Not knowing the realization of D in advance is more constraining, and the online optimum needs to satisfy the tighter constraints (16). The two LPs are equivalent in the special case where D deterministically equals T.

C.3. Algorithm and results based on the conditional LP

Given $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ and the thought experiment of conditioning on $D = T$, our algorithm for CORREL ∩ RAND is quite simple, and described in Algorithm 3.

Algorithm 3 Algorithm for CORREL ∩ RAND

Solve $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$, letting $(y^t_{i,j})_{t\in[T],i\in[n],j\in[m]}$ denote an optimal solution
$\mathsf{inv}[i] = k_i$ for all $i \in [n]$ ▷ starting inventory of resource i
for $t = 1, \ldots, D$ **do** ▷ realization D unknown to algorithm
  Let j denote type of query t ▷ drawn independently from probability vector $(p_j)_{j\in[m]}$
  Choose i independently from probability vector $\left(\frac{y^t_{i,j}}{p_j}\right)_{i\in[n]}$ ▷ $\sum_{i=1}^n \frac{y^t_{i,j}}{p_j} \le 1$ by (17)
  (set i to $\perp$ with probability $1 - \sum_{i=1}^n \frac{y^t_{i,j}}{p_j}$)
  if $i \neq \perp$ and $\mathsf{Acc}_{i,t}(\mathsf{inv}[i]) = 1$ **then** ▷ $\mathsf{Acc}_{i,t}(\mathsf{inv}[i])$ is random bit for whether i accepts
    Match the query to resource i, collecting reward $r_{i,j}$
    $\mathsf{inv}[i] = \mathsf{inv}[i] - 1$
  end if
end for

In Algorithm 3, $\mathsf{Acc}_{i,t}(\mathsf{inv}[i])$ is a random bit returned by an Online Contention Resolution Scheme (OCRS) for resource i, indicating whether to accept a query at time t if there are $\mathsf{inv}[i]$ units left. It does not discriminate based on the type j of the query, and the OCRS cannot accept if $\mathsf{inv}[i] = 0$. The purpose of an OCRS is to guarantee that the probability of accepting a query, at any time t, is always at least some constant $\gamma > 0$. Intuitively, an OCRS rejects early arrivals with some probability so that inventory is left for late arrivals; these probabilities are calibrated so that all arrivals have at least probability γ of being accepted. In particular, our OCRS-based algorithm is adaptive to the unfolding of the sequence of queries. The largest achievable value of γ then yields our desired approximation ratio.

The application of OCRS can be made precise by considering an equivalent, alternative world in which time t always runs until the maximum of T, but only rewards collected in times $t \le D$ are counted. From the perspective of any resource $i \in [n]$, it will be chosen by Algorithm 3 at each time $t \in [T]$ with probability

$$\sum_{j=1}^m p_j \frac{y^t_{i,j}}{p_j} = \sum_{j=1}^m y^t_{i,j}, \tag{19}$$

in which case we say that i is *active* at time t. Moreover, we have the following properties:

(Serial independence) The events $\{i \textit{ is active in time } t\}$ are mutually independent across $t \in [T]$, noting that the realization of D does not induce any serial correlation in the alternative world where the horizon always runs until time T;

(Feasibility in expectation) For each resource $i \in [n]$, the expected number of active queries, equal to $\sum_{t=1}^T \sum_{j=1}^m y^t_{i,j}$ by (19), is at most k_i, as imposed by our LP constraint (16).

These are the two conditions required to utilize OCRSes devised in previous literature (Jiang et al. 2022). Each resource i runs a separate OCRS that prescribes random bits $\mathsf{Acc}_{i,t}(\mathsf{inv}[i])$, satisfying $\mathsf{Acc}_{i,t}(0)=0$ and guaranteeing that $\Pr[\mathsf{Acc}_{i,t}(\mathsf{inv}[i])=1]\geq\gamma^*_{k_i}$ for all t. Here, $\gamma^*_{k_i}$ is a constant that depends on the starting inventory k_i of resource i, and the probability is over the randomness in the inventory state $\mathsf{inv}[i]$ at time t. Consequently, we leverage the OCRS guarantees for each combination of $(i,j,t)\in[n]\times[m]\times[T]$, noting that a reward is collected if: (i) at time t, query type j is drawn and resource i is chosen by Algorithm 3; (ii) $D\geq t$; and (iii) $\mathsf{Acc}_{i,t}(\mathsf{inv}[i])=1$. Because (i) and (ii) are independent from everything else, we obtain

$$\mathsf{ALG}(\mathcal{I})\geq\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{t=1}^{T}r_{i,j}y^t_{i,j}\Pr[D\geq t]\gamma^*_{k_i}\geq\gamma^*_{\min_i k_i}\cdot\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{t=1}^{T}r_{i,j}y^t_{i,j}\Pr[D\geq t]\right)=\gamma^*_{\min_i k_i}\cdot\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})\ .$$

To explain the second inequality, $\gamma^*_{k_i}$ is the best-possible OCRS guarantee for randomly rationing a resource with k_i units. This constant has been characterized in Jiang et al. (2022), and shown to be increasing in k_i. Therefore, $\gamma^*_{k_i}\geq\gamma^*_{\min_i k_i}$ for every $i\in[m]$, resulting in the following theorem.

Theorem 4. *For the* Correl ∩ Rand *model, Algorithm 3 is a polynomial-time online algorithm that is* γ^*_k*-approximate, satisfying* $\mathsf{ALG}(\mathcal{I})\geq\gamma^*_k\cdot\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})\geq\gamma^*_k\cdot\mathsf{OPT}(\mathcal{I})$ *for every instance* $\mathcal{I}$ *in which all resources* $i\in[m]$ *have a starting inventory* $k_i\geq k$.

We note that $\gamma^*_1=1/2$, and γ^*_k is greater than an earlier lower bound of $1-1/\sqrt{k+3}$ proved in Alaei (2014) for all $k>1$. This shows that Algorithm 3 is asymptotically optimal for Correl ∩ Rand as $k\to\infty$. Given the connection between Correl ∩ Rand and the stochastic horizon setting, Theorem 4 is also applicable in that setting. In fact, our conditional LP and OCRS reduction can be extended in two ways: (1) to a nonstationary version of the stochastic horizon setting, and (2) to Network Revenue Management where queries are matched to products that consume multiple resources. These extensions are presented in Appendix E.

We conclude with two negative results, both only requiring a single unit of a single resource.

Proposition 3. *For the* Correl ∩ Rand *model with a single unit of a single resource, any algorithm that accepts the first query above a fixed threshold has an approximation ratio of zero.*

Proposition 3 is proved in Appendix C.6. Intuitively, static threshold policies perform poorly because they do not increase their thresholds in rare occurrences when the horizon "survives" past unlikely cutoffs. By contrast, our algorithm uses $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$, which prescribes different matching probabilities $y^t_{i,j}$ and thus prescribes thresholds that are adapted to the time t. Proposition 3 also provides a noteworthy contrast to the single-item prophet inequality setting, in which a fixed threshold algorithm can be 1/2-approximate (Samuel-Cahn 1984). This suggests that, despite the simplicity of our analysis above, the Correl ∩ Rand and SI models are definitely not equivalent and analyzing the former is crucially dependent on our conditional LP.

Proposition 4. *For the* Correl ∩ Rand *model, we have* $\inf_{\mathcal{I}\in\text{Correl}\cap\text{Rand}}\dfrac{\mathsf{OPT}(\mathcal{I})}{\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})}\leq\dfrac{1}{2}$.

Finally, Proposition 4 is proved in Appendix C.7. It shows that the approximation ratio of 1/2 achieved in Theorem 4 (where $\gamma^*_1=1$) is tight relative to $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$, when $k=1$. We note that this requires constructing new hard instances because we measure performance against a less powerful benchmark than the offline optimum, and allow for arbitrary online algorithms.

C.4. Formal definition of $\mathsf{OPT}(\mathcal{I})$

We specify $\mathsf{OPT}(\mathcal{I})$, defined as the expected total rewards collected by an optimal policy found via computationally unconstrained dynamic programming, in the CORREL ∩ RAND model of Appendix C. In that section of the paper, we compare our algorithm's performance against $\mathsf{OPT}(\mathcal{I})$ instead of $\mathsf{OFF}(\mathcal{I})$. For every time $t \in [T]$ and remaining inventory levels $I_1, \ldots, I_n$ of each resource $i \in [n]$, let $J_t(I_1, \ldots, I_n)$ denote the value-to-go with remaining inventories $I_1, \ldots, I_n$ upon reaching the end of time $t-1$ (i.e., knowing that $D \geq t-1$ but not yet knowing whether $D \geq t$). Here, the value function $J_{T+1}(I_1, \ldots, I_n)$ is understood to to be 0 for all vectors $(I_1, \ldots, I_n)$, and $J_t(0, \ldots, 0)$ is understood to be 0 for all t. The Bellman equation then tell us that

$$J_t(I_1, \ldots, I_n) = \Pr[D \geq t | D \geq t-1] \sum_{j=1}^{m} p_{t,j} \max_{i:k_i>0} \left(r_{i,j} + J_{t+1}(I_1, \ldots, I_{i-1}, I_i - 1, I_{i+1}, \ldots, I_n) \right) \quad \forall t \in [T], (I_1, \ldots, I_n)$$

because conditional on $D \geq t$ and type j arriving next, we should match to the resource i that maximizes the immediate reward $r_{i,j}$ plus the value-to-go J_{t+1} with the i'th entry of the inventory vector being decremented by one unit. With this definition at hand, $\mathsf{OPT}(\mathcal{I})$ simply corresponds to $J_1(k_1, \ldots, k_n)$, where $k_1, \ldots, k_n$ are the starting inventory levels of $\mathcal{I}$.

C.5. Proof of Lemma 3

Fix any online algorithm, and on a sample path of its execution, let $X_{i,j}^t \in \{0,1\}$ be the indicator random variable for a query of type j being matched to resource i in time t. Let $Q_j^t \in \{0,1\}$ be the indicator random variable for a query at step t having type j, where we note that $Q^t := \sum_j Q_j^t \leq 1$, with $Q^t = 0$ whenever the total demand D is less than t.

On any sample path, the reward collected by the algorithm is $\sum_{i,j,t} r_{i,j} X_{i,j}^t$. Meanwhile, the constraint $\sum_{j,t} X_{i,j}^t \leq k_i$ must be satisfied for every resource i, since i can be matched at most k_i times; and the constraint $\sum_i X_{i,j}^t \leq Q_j^t$ must be satisfied for every j and T, since a query at step t of type j can be matched to at most one i, and only if $Q_j^t = 1$. Taking an expectation over *only the sample paths where* $Q_T = 1$, i.e., paths where D realized to its maximum value of T, we have

$$\sum_{j=1}^{m} \sum_{t=1}^{T} \mathbb{E}\left[X_{i,j}^t \middle| Q_T = 1 \right] \leq k_i \qquad \forall i \in [n]$$

$$\sum_{i=1}^{n} \mathbb{E}\left[X_{i,j}^t \middle| Q_T = 1 \right] \leq p_{t,j} \qquad \forall j \in [m], \forall t \in [T]$$

noting that $\mathbb{E}[Q_j^t] = p_{t,j}$. Now, we derive for any i, j, and t that

$$\mathbb{E}\left[X_{i,j}^t \middle| Q_T = 1 \right] \;=\; \mathbb{E}\left[X_{i,j}^t \middle| Q_t = 1 \right] \;=\; \frac{\mathbb{E}[X_{i,j}^t Q_t]}{\mathbb{E}[Q_t]} \;=\; \frac{\mathbb{E}[X_{i,j}^t]}{\Pr[D \geq t]}\,, \tag{20}$$

where the first equality holds because an online algorithm's decision $X_{i,j}^t$ at time t cannot distinguish between $Q_t = 1$ and the stronger future event that $Q_T = 1$, and the final equality holds because $X_{i,j}^t = 1$ implies $Q_t = 1$ while $\mathbb{E}[Q_t] = \Pr[D \geq t]$ by definition. Applying (20), the expected reward collected by the algorithm is $\mathbb{E}[\sum_{i,j,t} r_{i,j} X_{i,j}^t] = \sum_{i,j,t} r_{i,j} \mathbb{E}[X_{i,j}^t | Q_T = 1] \cdot \Pr[D \geq t]$.

Letting $y_{i,j}^t = \mathbb{E}[X_{i,j}^t | Q_T = 1]$ for all i, t, and j, we see that this forms a feasible solution to $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ with objective value equal to the expected reward collected by the algorithm. Since any online algorithm corresponds to such a feasible solution to $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$, the optimal objective value of $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ can be no less than the expected reward of any online algorithm, thereby completing the proof of Lemma 3.

C.6. Proof of Proposition 3

Let $n=1$ with $k_1=1$. Fix a large T and a small $\varepsilon>0$. Let D be distributed over $\{1,\dots,T\}$ as follows: for any $t=1,\dots,T-1$, the survival rate is $\Pr[D>t|D\geq t]=\varepsilon$. Types are defined so that conditional on any time $t\in[T]$ occurring, the reward for matching with the resource is $1/\varepsilon^t$ with probability ε, and $1/\varepsilon^{t-1}$ with probability $1-1/\varepsilon$.

We claim that $\mathsf{OPT}(\mathcal{I})\geq T-O(\varepsilon)$ by considering the following online algorithm. In any time t, it only accepts the query if the reward is the larger realization of $1/\varepsilon^t$ (which occurs with probability ε). Such an algorithm will accept a query in time t if $D\geq t$, which occurs with probability ε^{t-1}; conditional on this, no query is accepted before time t, which occurs with probability $(1-\varepsilon)^{t-1}$. Therefore,

$$\begin{aligned}\mathsf{ALG}(\mathcal{I}) &= \sum_{t=1}^{T}\varepsilon^{t-1}(1-\varepsilon)^{t-1}\varepsilon\cdot\frac{1}{\varepsilon^t}\\ &\geq T\cdot(1-\varepsilon)^T\ .\end{aligned}$$

Now consider any policy that sets a fixed threshold of $1/\varepsilon^t$, for some $t\in\{0,\dots,T\}$. This policy can only collect reward in at most 2 times, t or $t+1$. The maximum reward that can be collected in any time t (assuming there is available inventory, and accepting either realization $1/\varepsilon^t$ or $1/\varepsilon^{t-1}$) is

$$\Pr[D\geq t]\cdot\left(\varepsilon\cdot\frac{1}{\varepsilon^t}+(1-\varepsilon)\frac{1}{\varepsilon^{t-1}}\right)=\varepsilon^{t-1}\left(\varepsilon\cdot\frac{1}{\varepsilon^t}+(1-\varepsilon)\frac{1}{\varepsilon^{t-1}}\right)\leq 2\ .$$

Therefore, such a policy has expected reward upper-bounded by 4, which is an arbitrarily small fraction of $T\cdot(1-\varepsilon)^T$ as T tends to ∞ and ε tends to 0.

C.7. Proof of Proposition 4

Let $n=1$ and $k_1=1$. Fix a large integer N. Let $m=2$, with the type rewards given by $r_{1,1}=1$ and $r_{1,2}=N^2$. Let the distribution for the total demand D be $D=1$ with probability $1-1/N$, and $D=1+N^2$ with probability $1/N$. Each query draws an IID type that is 1 with probability $p_1:=1-1/N^3$ and 2 with probability $p_2:=1/N^3$.

The following is a feasible solution with respect to $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$: set $y^1_{1,1}=1-(1+N^2)/N^3<p_1$ and $y^1_{1,2}=1/N^3=p_2$; set $y^t_{1,1}=0$ and $y^t_{1,2}=1/N^3$ for $t=2,\dots,1+N^2$. Intuitively, this fractional solution accepts both types of queries when $t=1$, and only accepts queries of type 2 when the time horizon D "survives" past $t=1$. Constraint (17) is satisfied by construction, and constraint (16) can be verified because its left-hand side equals $1-(1+N^2)/N^3+(1+N^2)\cdot 1/N^3=1$. Now, the objective value is

$$\begin{aligned}1-\frac{1+N^2}{N^3}+N^2\left(\frac{1}{N^3}+\Pr[D\geq 2]\cdot N^2\cdot\frac{1}{N^3}\right) &= 1-\frac{1}{N^3}+N^2\left(\frac{1}{N}N^2\cdot\frac{1}{N^3}\right)\\ &= 2-O\left(\frac{1}{N^3}\right)\ .\end{aligned}$$

Meanwhile, consider any online algorithm. If its plan is to accept type 1 in time 1, then its expected reward is $1-1/N^3+N^2\cdot 1/N^3=1+O(1/N)$. On the other hand, if its plan is to reject type 1 in time 1, then its expected reward is at most $N^2\cdot(1+1/N\cdot N^2)\cdot 1/N^3$ which is also $1+O(1/N)$. Taking $N\to\infty$ completes the proof.

Appendix D: Extension to Sampling-Based Setting

Our result for INDEP extends to the setting where the decision-maker only has access to IID samples of the demand vector rather than its exact distribution. We do not analyze the CORREL model in this sampling-based setting, leaving the robustness of Theorem 4 to sampling error as an open question. We note that CORREL is more similar to the SI model, for which sample-based competitive algorithms are known. The key difference is the distribution of the stochastic horizon D, which intuitively should be easier to learn from data than the marginal distributions for each demand type in INDEP.

Here, we suppose that the distribution of $\mathbf{D}$ is unknown but can be estimated from a sample of N IID draws $\hat{\mathbf{D}} = (\hat{\mathbf{D}}_1, \ldots, \hat{\mathbf{D}}_N)$ from the distribution of $\mathbf{D}$. We present sample complexity results for INDEP, showing that for N polynomially large, we can incur a small loss $O(\varepsilon)$ in our performance guarantee.

We can leverage existing results on the generalization error of hypothesis classes under product form distributions to obtain additive approximation errors. For every demand type $j \in [m]$, we define $\hat{E}_j$ as the empirical distribution over $\hat{D}_{1,j}, \hat{D}_{2,j}, \ldots, \hat{D}_{N,j}$. Here, we assume that the rewards are normalized such that $\max_{i\in[n],j\in[m]} r_{i,j} = 1$ and $k_i = 1$ for every resource $i \in [n]$.

PROPOSITION 5. *There exists a universal constant $C > 0$ such that for all $N \geq C\frac{n^3 m}{\varepsilon^2}\log\frac{1}{\delta}$, running Algorithm 1 with respect to the product-form empirical distribution $\hat{\mathbb{E}} = \hat{E}_1 \times \cdots \times \hat{E}_m$ achieves $\inf_{\mathcal{I}\in \text{INDEP}\cap\text{ADV}} \mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}(\mathcal{I}) \geq \frac{1}{2}\cdot \mathsf{LP}^{\mathsf{trunc}}(\mathcal{I}) - O(\varepsilon)$ with probability at least $1-\delta$.*

This result follows from Theorem 5 of Guo et al. (2021), noting that the demand for each type j can be truncated to be at most n without loss (i.e., $\Pr[D_j \leq n] = 1$ for all $j \in [m]$). We maintain the assumption that $D_j \leq n$ with probability 1 for all $j \in [m]$ throughout this section.

To elaborate on how to establish Proposition 5, let $\mathcal{I}$ denote the instance with the true distribution, and $\hat{\mathcal{I}}$ denote the instance with the same rewards $(r_{i,j})_{i\in[n],j\in[m]}$ and starting inventories $(k_i)_{i\in[n]}$ but using the empirical distribution instead. Our algorithm trains an online policy (a fixed prescription of how to match queries on-the-fly) based on $\hat{\mathcal{I}}$, which will satisfy $\mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}(\hat{\mathcal{I}}) \geq \frac{1}{2}\cdot\mathsf{LP}^{\mathsf{trunc}}(\hat{\mathcal{I}})$. Here, we use $\mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}$ instead of ALG to emphasize that our actual policy is trained on the empirical distribution. However, what the algorithm actually earns is $\mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}$, which we wish compare to $\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$. Fortunately, the result of Guo et al. (2021) says that with sufficiently many samples N, we can uniformly guarantee for any real-valued function of a distribution that its evaluation on the true vs. empirical distributions differ by at most a small $\varepsilon > 0$. Because $\mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}(\cdot)$ and $\mathsf{LP}^{\mathsf{trunc}}(\cdot)$ can both be viewed as real-valued functions that depend on the distribution in the instance, we apply the result of Guo et al. (2021) twice to get that

$$\begin{aligned}
\mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}(\mathcal{I}) &\geq \mathsf{ALG}_{\hat{\mathbb{E}}(\mathcal{I})}(\hat{\mathcal{I}}) - \varepsilon \\
&\geq \frac{1}{2}\cdot\mathsf{LP}^{\mathsf{trunc}}(\hat{\mathcal{I}}) - \varepsilon \\
&\geq \frac{1}{2}\cdot\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I}) - 2\varepsilon.
\end{aligned}$$

The required number of samples N scales with the square of the range of the functions, which in this case is $[0, n]$, under the normalization of the rewards. This explains why the sample complexity is $O(\frac{n^3 m}{\varepsilon^2}\log\frac{1}{\delta})$ instead of $O(\frac{nm}{\varepsilon^2}\log\frac{1}{\delta})$ as quoted in Guo et al. (2021, Thm 5).

A drawback of Proposition 5 is that the loss $O(\varepsilon)$ might be large in the regime where $\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$ is small. Thus, we propose a data-driven algorithm to achieve an $O(\varepsilon)$-multiplicative error. The following result holds even if our algorithm does not observe any type $j_0 \in [m]$ without any arrival in the sample $\sum_{t=1}^{N} \hat{D}_{t,j_0} = 0$, and it needs not know their corresponding rewards $(r_{i,j_0})_{i\in[n]}$. That said, our sample complexity bound implicitly assumes that we know an upper bound on the total number of customer types m.

THEOREM 5. *There exist a polynomial-time algorithm and a universal constant $C > 0$ such that for all $N \geq C\frac{nm}{\varepsilon^3}\log\frac{nm}{\delta}$, $\inf_{\mathcal{I}\in\textsc{Indep}\cap\textsc{Adv}} \mathsf{ALG}_{\hat{\mathbf{D}}\sim\mathbf{D}^N(\mathcal{I})}(\mathcal{I}) \geq (1-\varepsilon)\cdot\frac{1}{2}\cdot\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$ with probability at least $1-\delta$, where $\mathsf{ALG}_{\hat{\mathbf{D}}\sim\mathbf{D}^N(\mathcal{I})}(\mathcal{I})$ stands for the expected rewards in $\mathcal{I}$ for each sample instantiation $\hat{\mathbf{D}} \sim \mathbf{D}^N(\mathcal{I})$.*

The remainder of the section is devoted to proving Theorem 5. At a high level, the algorithm calls our original $\mathsf{LP}^{\mathsf{trunc}}(\hat{\mathcal{I}})$ and lossless rounding for a specific input instance $\hat{\mathcal{I}}$, estimated using the sample $\hat{\mathbf{D}}$, but it gives priority to matching infrequent types whose demand distribution cannot be accurately inferred from the sample. In what follows, we hide the dependence on $\mathcal{I}$ whenever it is clear which instance we refer to and only indicate which demand distribution is used to evaluate the outcomes. We start by stating a few basic technical claims. We denote by $\mathbf{D} \succeq \boldsymbol{E}$ the (first-order) stochastic dominance order $\Pr[D_j \geq \ell] \geq \Pr[E_j \geq \ell]$ for all $j \in [m]$ and $\ell \geq 1$.

CLAIM 3 **(Dominated distribution)**. *For every product-form distribution $\mathcal{D}$, having access to a sample of $N \geq \frac{3nm}{\varepsilon^3}\log\frac{2nm}{\delta}$ independent draws from that distribution $(\hat{\mathbf{D}}_1,\ldots,\hat{\mathbf{D}}_N) \sim \mathbf{D}^N$, we can construct in polynomial time a product-form distribution $\boldsymbol{E}$ such that, with probability at least $1-\delta$, we have $\mathbf{D} \succeq \boldsymbol{E}$ and $\Pr[E_j \geq \ell] \geq (1-2\varepsilon)\Pr[D_j \geq \ell]$ for all $j \in [m], \ell \in [n]$ for which $\Pr[D_j \geq \ell] \geq \frac{\varepsilon}{mn}$.*

Proof. Fix $(j,\ell) \in [m]\times[n]$ for which $\Pr[D_j \geq \ell] \geq \frac{\varepsilon}{mn}$. We invoke the multiplicative Chernoff bound, using the shorthand $p_{j,\ell} = \Pr[D_j \geq \ell]$,

$$\Pr_{\hat{\mathbf{D}}_1,\ldots,\hat{\mathbf{D}}_N\sim\mathbf{D}^N}\left[\left|\frac{1}{N}\cdot\sum_{t=1}^{N}\mathbb{I}\left[\hat{D}_{t,j}\geq\ell\right]-p_{j,\ell}\right|\geq\varepsilon p_{j,\ell}\right]\leq 2e^{-\frac{\varepsilon^2 N p_{j,\ell}}{3}}\ . \tag{21}$$

It follows that, by defining $\hat{p}_{j,\ell} = (1-\varepsilon)\cdot\frac{1}{N}\cdot\sum_{t=1}^{N}\mathbb{I}[\hat{D}_{t,j}\geq\ell]$, we obtain

$$\begin{aligned}&\Pr_{\hat{\mathbf{D}}_1,\ldots,\hat{\mathbf{D}}_N\sim\mathbf{D}^N}\left[\hat{p}_{j,\ell}\notin((1-\varepsilon)^2 p_{j,\ell},(1+\varepsilon)(1-\varepsilon)p_{j,\ell})\right]\leq\frac{\delta}{nm}\\ \Longrightarrow\ &\Pr_{\hat{\mathbf{D}}_1,\ldots,\hat{\mathbf{D}}_N\sim\mathbf{D}^N}\left[\hat{p}_{j,\ell}\notin[(1-2\varepsilon)p_{j,\ell},p_{j,\ell}]\right]\leq\frac{\delta}{nm}\end{aligned} \tag{22}$$

where we use inequality (21), and the hypotheses $p_{j,\ell} = \Pr[D_j \geq \ell] \geq \frac{\varepsilon}{mn}$ and $N \geq \frac{3nm}{\varepsilon^3}\log\frac{2nm}{\delta}$. Consequently, we define $\Pr[E_j \geq \ell] = \min\{\hat{p}_{j,\ell}, \Pr[E_j \geq \ell-1]\}$ iteratively over $\ell \in [1,n]$ with $\Pr[E_j \geq 0] = 1$ and $\Pr[E_j \geq n+1] = 0$. The desired claim follows from the union bound over (j,ℓ) and inequality (22). □

CLAIM 4 **(LP Sensitivity)**. *For all product-form distributions $\mathbf{D} \succeq \boldsymbol{E}$, and for all $\varepsilon > 0$,*

- *if $\Pr[D_j \geq \ell] - \Pr[E_j \geq \ell] \leq \varepsilon$ for all $j \in [m]$, $\ell \in [n]$, then $\mathsf{LP}^{\mathsf{trunc}}(\boldsymbol{E}) \geq \mathsf{LP}^{\mathsf{trunc}}(\mathbf{D}) - O(\varepsilon nm)$.*
- *if $\Pr[E_j \geq \ell] \geq (1-\varepsilon)\cdot\Pr[D_j \geq \ell]$ for all $j \in [m]$, $\ell \in [n]$, then $\mathsf{LP}^{\mathsf{trunc}}(\boldsymbol{E}) \geq (1-\varepsilon)\cdot\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$.*

CLAIM 5 **(Median prophet)**. *Suppose that $X_1, \ldots, X_n$ is a sequence of independent two-outcome random variables with rewards $0 = r_0 < r_1 < r_2 < \ldots < r_n$ and probabilities $\Pr[X_i = r_i] = p_i$ and $\Pr[X_i = 0] = 1 - p_i$ such that $\sum_{i=1}^n p_i \leq 1$. The unique (randomized) perturbed threshold τ_{ptb} such that $\tau_{\text{ptb}} \in \{r_i, r_{i+1}\}$ for some $i \in [0, n-1]$ and $\Pr[\max_{i\in[n]} X_i \geq \tau_{\text{ptb}}] = \frac{1}{2}$ yields the prophet inequality $\mathbb{E}[R(\tau_{\text{ptb}})] \geq \frac{1}{2}(\sum_{i=1}^n r_i p_i)$, where $R(\tau_{\text{ptb}})$ is equal to the first outcome above τ_{ptb}, if any, and otherwise to zero. Moreover, when running this static threshold rule with noisy inputs $\tilde{p}_1, \ldots, \tilde{p}_n$ instead of $p_1, \ldots, p_n$, where $\tilde{p}_i \geq (1-\varepsilon)p_i$ and $\sum_{i=1}^n \tilde{p}_i \leq \sum_{i=1}^n (1-\varepsilon)^{-1} p_i + \varepsilon$ for all $i \in [n]$ with an $\varepsilon \in (0, \frac{1}{2})$, the resulting perturbed threshold $\tilde{\tau}_{\text{ptb}}$ yields $\mathbb{E}[R(\tilde{\tau}_{\text{ptb}})] \geq \frac{(1-11\varepsilon)}{2} \cdot (\sum_{i=1}^n r_i p_i)$ and $\Pr[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}}] \leq \frac{(1+2\varepsilon)}{2}$.*

Proof. We prove the second claim, which implies the first one. By construction, we have $\Pr[\tilde{\tau}_{\text{ptb}} \geq \max_{i\in[n]} \tilde{X}_i] = 1/2$ for the sequence of independent modified random variables $\tilde{X}_i$ with $\Pr[\tilde{X}_i = r_i] = \tilde{p}_i$. In particular, there exists $i^* \in [0, n-1]$ such that $\Pr[\tilde{\tau}_{\text{ptb}} = r_{i^*}] = \delta > 0$ and $\Pr[\tilde{\tau}_{\text{ptb}} = r_{i^*+1}] = 1 - \delta$. Define $\delta_i = 0$ for $i \in [0, i^* - 1]$, $\delta_i = 1$ for $i \in [i^* + 1, n]$, and $\delta_{i^*} = \delta$. The calibration of the threshold parameters i^*, δ such that $\Pr[\tilde{\tau}_{\text{ptb}} \geq \max_{i\in[n]} \tilde{X}_i] = 1/2$ is equivalent to the identity $\sum_{i=1}^n \tilde{p}_i \cdot \delta_i \cdot \prod_{i'=1}^{i-1}(1 - \tilde{p}_{i'} \cdot \delta_{i'}) = \frac{1}{2}$. Defining $\varepsilon_i = \tilde{p}_i - (1-\varepsilon)^{-1} p_i$, it follows that

$$\begin{aligned}
\frac{1}{2} &\leq \sum_{i=1}^n ((1-\varepsilon)^{-1} p_i + \varepsilon_i) \cdot \delta_i \cdot \left(\prod_{i'=1}^{i-1} (1 - (1-\varepsilon) p_{i'} \cdot \delta_{i'}) \right) \\
&\leq 3\varepsilon + (1-\varepsilon)^{-1} \sum_{i=1}^n p_i \cdot \delta_i \cdot \left(\prod_{i'=1}^{i-1} e^{\log(1 - p_{i'} \cdot \delta_{i'}) + 2\varepsilon p_{i'} \delta_{i'}} \right) \\
&= 3\varepsilon + (1-\varepsilon)^{-1} e^{(\sum_{i=1}^n 2\varepsilon p_i \delta_i)} \Pr\left[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}} \right] \\
&\leq 3\varepsilon + (1-5\varepsilon)^{-1} \Pr\left[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}} \right] \qquad (23)
\end{aligned}$$

where in the first inequality we use $\tilde{p}_i \geq (1-\varepsilon)p_i$ for all $i \in [n]$. In the second inequality, we use the fact that $\varepsilon_i \geq -2\varepsilon p_i$ and thus $\sum_{i=1}^n [\varepsilon_i]^+ = \sum_{i=1}^n (\varepsilon_i + 2\varepsilon p_i) \leq 3\varepsilon$ by hypothesis (because $\sum_{i=1}^n \varepsilon_i \leq \sum_{i=1}^n \tilde{p}_i - \sum_{i=1}^n (1-\varepsilon)^{-1} p_i \leq \varepsilon$), and notice that $\log(1 - (1-\varepsilon)u) \leq \log(1-u) + \varepsilon u \frac{1}{1-u}$ with $u = p_{i'} \delta_{i'} \leq \frac{1}{2}$. In the last inequality, we use the fact that $\sum_{i=1}^n 2\varepsilon p_i \delta_i \leq 2\varepsilon$ and $\varepsilon \in (0, \frac{1}{2})$. Reciprocally, from $\tilde{p}_i \geq (1-\varepsilon)p_i$ for all $i \in [n]$, we obtain

$$\begin{aligned}
\frac{1}{2} &= \sum_{i=1}^n \tilde{p}_i \cdot \delta_i \cdot \left(\prod_{i'=1}^{i-1} (1 - \tilde{p}_{i'} \cdot \delta_{i'}) \right) \\
&\geq \sum_{i=1}^n (1-\varepsilon) p_i \cdot \delta_i \cdot \left(\prod_{i'=1}^{i-1} (1 - (1-\varepsilon) p_{i'} \cdot \delta_{i'}) \right) \\
&\geq (1-\varepsilon) \sum_{i=1}^n p_i \cdot \delta_i \cdot \left(\prod_{i'=1}^{i-1} (1 - p_{i'} \cdot \delta_{i'}) \right) \\
&\geq (1-\varepsilon) \Pr\left[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}} \right] , \qquad (24)
\end{aligned}$$

where the first inequality holds since $[\boldsymbol{x} \mapsto \sum_{i=1}^n x_i \cdot \delta_i \cdot (\prod_{i'=1}^{i-1}(1 - x_{i'} \cdot \delta_{i'}))]$ is non-decreasing in $\boldsymbol{x}$.

Now, we use the standard decomposition

$$\begin{aligned}
&\mathbb{E}\left[\min\left\{X_i : i \in [n], \mathbb{I}[X_i \geq \tilde{\tau}_{\text{ptb}}]\right\}\right] \\
&\quad \geq \Pr\left[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}} \right] \cdot r_{i^*} + \left(1 - \Pr\left[\max_{i\in[n]} X_i \geq \tilde{\tau}_{\text{ptb}} \right] \right) \sum_{i=1}^n \max\{0, r_i - r_{i^*}\} \cdot p_i
\end{aligned}$$

$$\geq \frac{1-11\varepsilon}{2} \cdot \left(\sum_{i=1}^{n} r_i p_i \right) ,$$

where the first inequality holds since $r_i > r_{i^*}$ implies $\delta_i = 1$, and the last inequality follows from the fact that $\sum_{i=1}^{n} p_i \leq 1$ and inequalities (23) and (24). $\square$

Data-driven algorithm. Fixing $\mathbf{D}$, we distinguish between frequent and infrequent queries with $\mathcal{T}_{\text{freq}} = \{(j,\ell) \in [m] \times [n] : \Pr[D_j \geq \ell] \geq \frac{\varepsilon}{mn}\}$. The first step of our algorithm is to use the sample $\hat{\mathbf{D}}$ to construct the product-form distribution $\boldsymbol{E} \preceq \boldsymbol{D}$ achieving the properties stated in Claim 3. Next, we determine the subset of frequent types $\hat{\mathcal{T}}_{\text{freq}} = \{(j,\ell) \in [m] \times [n] : \Pr[E_j \geq \ell] \geq (1-2\varepsilon) \cdot \frac{2\varepsilon}{mn}\}$ and infrequent types $\mathcal{T}_{\text{infreq}} = [m] \times [n] \setminus \hat{\mathcal{T}}_{\text{freq}}$. By Claim 3, with probability $1-\delta$, we have $\hat{\mathcal{T}}_{\text{freq}} \subseteq \mathcal{T}_{\text{freq}}$, $\Pr[D_j \geq \ell] \geq \Pr[E_j \geq \ell] \geq (1-2\varepsilon)\Pr[D_j \geq \ell]$ for all $(j,\ell) \in \hat{\mathcal{T}}_{\text{freq}}$, and $\Pr[D_j \geq \ell] \leq \frac{2\varepsilon}{mn}$ for all $(j,\ell) \in \hat{\mathcal{T}}_{\text{infreq}}$. Throughout the remainder of the analysis, we condition on the corresponding event and assume that these properties hold.

As a second step, we ignore infrequent arrivals and leverage Algorithm 1 with respect to the residual demand distribution. That is, we consider the modified demand distribution $\bar{\boldsymbol{E}}$ after we eliminate every infrequent query $(j,\ell) \in \hat{\mathcal{T}}_{\text{infreq}}$, i.e., $\Pr[\bar{E}_j \geq \ell] = \Pr[E_j \geq \ell]$ for every $(j,\ell) \in \hat{\mathcal{T}}_{\text{freq}}$ and $\Pr[\bar{E}_j \geq \ell] = 0$ for every $(j,\ell) \notin \hat{\mathcal{T}}_{\text{freq}}$ with renormalization $\Pr[\bar{E}_j = 0] = 1 - \sum_{\ell:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \Pr[\bar{E}_j \geq \ell]$. Consequently, let $\mathsf{ALG}_{\bar{\boldsymbol{E}}}$ denote the execution of Algorithm 1 by specifying the modified distribution $\bar{\boldsymbol{E}}$ as input and the median threshold rule. That is, $\mathsf{ALG}_{\bar{\boldsymbol{E}}}$ solves $\mathsf{LP}^{\mathsf{trunc}}(\bar{\boldsymbol{E}})$, then routes the arriving queries to resources using our lossless rounding Algorithm 2, and finally it runs the threshold rule of Claim 4 for each resource. The use of the threshold in Claim 4 marks a slight difference with Algorithm 1.

Lastly, as a third step, we consider the "augmented" algorithm $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ that deals with the arrivals of infrequent types, which are not expected to arrive as per $\bar{\boldsymbol{E}}$. Specifically, $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ takes the same matching decisions as $\mathsf{ALG}_{\bar{\boldsymbol{E}}}$ until an infrequent query $(j,\ell) \notin \hat{\mathcal{T}}_{\text{freq}}$ arrives, if any. From that point onwards, it matches all subsequent arrivals of type j greedily to the remaining resources and it no longer matches any query from other types. This completes the description of $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ as our data-driven algorithm for Theorem 5.

Analysis. The next claim relates the performance of $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ on the arrivals described by $\mathbf{D}$ to our LP benchmark, thereby completing the proof of Theorem 5.

LEMMA 4. $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}(\mathbf{D}) \geq \frac{(1-26\varepsilon)}{2}\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$.

To prove Lemma 4, we first lower bound the contributions of infrequent queries to the expected total rewards. For the purpose of analysis, we create a monotone coupling ω between the arrivals in $\bar{\boldsymbol{E}}$ and those in $\mathbf{D}$. Hence, we introduce $\zeta_{j,\ell,i}(\omega)$ as an indicator random variable such that $\zeta_{j,\ell,i}(\omega) = 1$ if the ℓ-th arrival of type j is routed to i by $\mathsf{ALG}_{\bar{\boldsymbol{E}}}$ on the arrivals described by $\bar{\boldsymbol{E}}(\omega)$. Similarly, we define $\zeta^{\uparrow}_{j,\ell,i}(\omega)$ as an indicator random variable such that $\zeta^{\uparrow}_{j,\ell,i}(\omega) = 1$ if the ℓ-th arrival of type j is routed to i by $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ on the arrivals described by $\mathbf{D}(\omega)$. Call $\mathcal{E}^{j}_{\text{infreq}}$ the event where at least one infrequent query of type j arrives by the end of the process. Because ω is a monotone coupling between $\bar{\boldsymbol{E}}$ and $\mathbf{D}$, and $\mathsf{ALG}_{\bar{\boldsymbol{E}}}, \mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ make the same decisions until reaching an infrequent query, we infer that for all $(j,\ell) \in \hat{\mathcal{T}}_{\text{freq}}$,

$$\zeta^{\uparrow}_{j,\ell,i}(\omega) \geq \left(1 - \sum_{j' \neq j} \mathbb{I}[\mathcal{E}^{j}_{\text{infreq}}]\right) \cdot \zeta_{j,\ell,i}(\omega) , \tag{25}$$

where we only sum over $j' \neq j$ because all infrequent queries of type j arrive after the frequent ones, and therefore, they cannot affect the routing decisions of frequent queries within the same type j. Now, observe

$$\sum_{j=1}^{m} \Pr\left[\mathcal{E}^j_{\text{infreq}}\right] = \sum_{(j,\ell)\in\hat{\mathcal{T}}_{\text{infreq}}} \Pr\left[D_j \geq \ell\right] \leq \sum_{(j,\ell)\in\mathcal{T}_{\text{infreq}}} \Pr\left[D_j \geq \ell\right] \leq 2\varepsilon \ , \tag{26}$$

where the last inequality follows from the definition of $\mathcal{T}_{\text{infreq}}$. By combining inequalities (25) and (26), we infer from the independence across types that $\mathbb{E}[\zeta^{\uparrow}_{j,\ell,i}(\omega)] \geq (1-2\varepsilon)\mathbb{E}[\zeta_{j,\ell,i}(\omega)]$. Reciprocally, we observe that $\Pr[E_j \geq \ell] \geq (1-2\varepsilon)\Pr[D_j \geq \ell]$ by construction for each $(j,\ell) \in \hat{\mathcal{T}}_{\text{freq}}$, implying that, for any such frequent query,

$$\mathbb{E}\left[\zeta_{j,\ell,i}(\omega)\right] \geq (1-2\varepsilon)\mathbb{E}\left[\zeta^{\uparrow}_{j,\ell,i}(\omega)\right] \ . \tag{27}$$

Now, denote by $\xi^{\uparrow}_{j,\ell,i}(\omega)$ the indicator of whether (j,ℓ) is eventually matched to resource i by $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ using the threshold rule in Claim 5. We obtain

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\substack{\ell\geq 1:\\ (j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}}} r_{i,j}\xi^{\uparrow}_{j,\ell,i}(\omega)\right] &\geq \frac{(1-22\varepsilon)}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell\geq 1} r_{i,j}\mathbb{E}\left[\zeta^{\uparrow}_{j,\ell,i}(\omega)\right]\right) \\
&\geq \frac{(1-24\varepsilon)}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell\geq 1} r_{i,j}\mathbb{E}\left[\zeta_{j,\ell,i}(\omega)\right]\right) \\
&= \frac{(1-24\varepsilon)}{2}\mathsf{LP}^{\mathsf{trunc}}(\bar{\boldsymbol{E}}) \\
&\geq \frac{(1-26\varepsilon)}{2}\mathsf{LP}^{\mathsf{trunc}}(\bar{\mathbf{D}}) \ ,
\end{aligned}$$

where in the first inequality, we apply Claim 5 to each resource, noting that we indeed route each query type j with probability $p_j = \sum_{\ell\geq 1:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \mathbb{E}[\zeta^{\uparrow}_{j,\ell,i}(\omega)]$, while our algorithm believes the true rate to be $\tilde{p}_j = \sum_{\ell\geq 1:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \mathbb{E}[\zeta_{j,\ell,i}(\omega)]$, which satisfies $\mathbb{E}[\zeta^{\uparrow}_{j,\ell,i}(\omega)] \geq (1-2\varepsilon)\mathbb{E}[\zeta_{j,\ell,i}(\omega)]$ as previously noted, and $\sum_{(j,\ell)} \mathbb{E}[\zeta^{\uparrow}_{j,\ell,i}(\omega)] \leq \sum_{(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \mathbb{E}[\zeta^{\uparrow}_{j,\ell,i}(\omega)] + \Pr[\bigcup_{j=1}^{m} \mathcal{E}^j_{\text{infreq}}] \leq (1-2\varepsilon)^{-1}\tilde{p}_j + 2\varepsilon$, where the last inequality follows from (26) and (27). Note that $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ does not alter the fact that the routed infrequent queries are independent from the perspective of each resource. The first equality follows from Lemma 2 using the fact that $\mathsf{ALG}_{\bar{\boldsymbol{E}}}$ implements our lossless rounding with respect to an optimal solution of $\mathsf{LP}^{\mathsf{trunc}}(\bar{\boldsymbol{E}})$. The last inequality follows from Claim 4, where $\bar{\mathbf{D}}$ is defined be to $\mathbf{D}$ with infrequent types removed.

The remainder of the proof looks at the contributions of infrequent queries to the expected rewards. Here, we first consider the natural coupling ω between lossless routing decisions for $\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$ and the routing decisions of our algorithm $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ on $\mathbf{D}$. Specifically, we implement Algorithm 2 with respect to an optimal solution of $\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$ and denote by $\delta_{j,\ell,i}(\omega) \in \{0,1\}$ the corresponding routing decisions for every query (j,ℓ) and resource i. In parallel, we denote by $\zeta^{\uparrow}_{j,\ell,i}(\omega)$ our algorithm's routing decisions for frequent queries, as described above, on the same sequence of arrivals. Next, we "realign" the routing decisions to concentrate their differences on infrequent queries. To be more specific, define $\delta^{\text{swap}}_{j,\ell,i}(\omega)$ as a new indicator variable that captures the following modified routing decision: at the end of the horizon, for each fixed $j \in [m]$, we consider the set of resources $i \in C^{\uparrow}_j(\omega)$ that have been routed a *frequent* arrival of type j by our algorithm $\mathsf{ALG}^{\uparrow}_{\bar{\boldsymbol{E}}}$ and the subset of resources $i \in C_j(\omega) \subseteq C^{\uparrow}_j(\omega)$ that were *also* routed any arrival of type j by the lossless

rounding of $\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$, i.e., $i \in C_j^{\uparrow}(\omega)$ if and only if $\sum_{\ell \geq 1:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \zeta^{\uparrow}_{j,\ell,i}(\omega) = 1$, and $i \in C_j(\omega)$ if and only if $\sum_{\ell\geq 1} \delta_{j,\ell,i}(\omega) = \sum_{\ell \geq 1:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \zeta^{\uparrow}_{j,\ell,i}(\omega) = 1$. Then, we construct an alternative routing strategy for $\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$, denoted by $\delta^{\text{swap}}_{j,\ell,i}(\omega) = \delta_{j,\pi_j(\ell),i}(\omega)$, where π_j is a permutation ensuring that all common resources in $C_j(\omega)$ are rerouted frequent queries but the set of resources with a routed query is unchanged, i.e., π_j swaps each infrequent arrival ℓ such that $\delta_{j,\ell,i} = 1$ for some $i \in C_j(\omega)$ with an earlier ℓ' such that $(j,\ell') \in \hat{\mathcal{T}}_{\text{freq}}$ but $\delta_{j,\ell',i'} = 1$ for some $i' \notin C_j(\omega)$. Because both processes are coupled by ω and thus there is the same realized demand per type, this swapping is always feasible. Importantly, the swapped routing might not be implementable as an online algorithm because defining π_j requires knowing the entire sequence of arrivals, but this won't matter for analysis. The main properties of the alternative routing are stated next.

Claim 6. *The swapped routing of* $\mathsf{LP}(\mathbf{D})$ *on* ω *satisfies:*

1. $\mathbb{E}[\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell\geq 1} r_{i,j}\delta^{\text{swap}}_{j,\ell,i}(\omega)] = \mathsf{LP}^{\mathsf{trunc}}(\mathbf{D})$.
2. *Let* $x_{i,j} = \mathbb{E}[\sum_{\ell\geq 1:(j,\ell)\in\hat{\mathcal{T}}_{\text{freq}}} \delta^{\text{swap}}_{j,\ell,i}(\omega)]$. *Then,* $(x_{i,j})_{i\in[n],j\in[m]}$ *is feasible in* $\mathsf{LP}^{\mathsf{trunc}}(\bar{\mathbf{D}})$.

Property 1 immediately follows from the fact swapping queries of the same type does not alter the total number of routing decisions per resource and query type, and thus $\sum_{\ell\geq 1}\delta^{\text{swap}}_{j,\ell,i}(\omega) = \sum_{\ell\geq 1}\delta_{j,\pi_j(\ell),i}(\omega)$ on every realization of ω. Property 2 follows from the fact that the fractional flow $(x_{i,j})_{i\in[n],j\in[m]}$ is the expectation of feasible matchings in the random graph between frequent queries in $\mathbf{D}$ and resources in $[n]$; in particular, it satisfies inequalities that, in expectation, correspond to the truncated LP's constraints in $\mathsf{LP}^{\mathsf{trunc}}(\bar{\mathbf{D}})$.

With this coupling at hand, we are ready to compare the expected reward contributions of infrequent types in $\mathsf{ALG}^{\uparrow}_{\boldsymbol{E}}$ to those described by $\delta^{\text{swap}}_{j,\ell,i}(\omega)$. Upon the first infrequent arrival (j,ℓ_j) of type j, let $i_1^j, i_j^2, \ldots, i_j^{U_j}$ be the random sequence of resources in $[n]\setminus C_j^{\uparrow}(\omega)$ with $U_j = n - \ell_j + 1$ rearranged by decreasing order of revenue $r_{i_1^j} \geq r_{i_j^2} \geq \ldots \geq r_{i_j^U}$. Note that ℓ_j and U_j are deterministic. Then, it is clear that

$$\sum_{i\in[n]}\sum_{\ell\geq\ell_j} r_{i,j}\delta^{\text{swap}}_{j,\ell,i}(\omega) \leq \sum_{u=1}^{U_j} \mathbb{I}[D_j \geq u + \ell_j - 1]\cdot r_{i_u^j}\ , \tag{28}$$

where the inequality holds because, by construction, all resources in $C_j(\omega)$ were routed a frequent query by $\delta^{\text{swap}}_{j,\ell,i}(\omega)$, and thus, they cannot be routed an infrequent query later. Moreover, all resources in $i \in C_j^{\uparrow}(\omega)\setminus C_j(\omega)$ satisfy $\sum_{\ell\geq\ell_j}\delta^{\text{swap}}_{j,\ell,i}(\omega) \leq \sum_{\ell\geq 1}\delta_{j,\ell,i}(\omega) = 0$ by the definition of $C_j(\omega)$. Therefore, out of the remaining resources $i \in [n]\setminus C_j^{\uparrow}(\omega)$, the best we can do is to select them by decreasing order of their rewards. We now argue that the contributions of infrequent queries matched by $\mathsf{ALG}^{\uparrow}_{\boldsymbol{E}}$ to the expected rewards generate a fraction at least $\frac{(1-2\varepsilon)}{2}$ of the righthand side in inequality (28). By extension of our definition, let $\xi^{\uparrow}_{j,\ell,i}(\omega)$ be the indicator for whether a query (j,ℓ) gets *matched* to resource i by $\mathsf{ALG}^{\uparrow}_{\boldsymbol{E}}$. Recall that $\mathcal{E}^j_{\text{infreq}}$ is the event where at least one infrequent query of type j arrives. Denote by $\mathcal{F}^j_{\text{infreq}} = \mathcal{E}^j_{\text{infreq}} \cap (\bigcap_{j'\neq j}\bar{\mathcal{E}}^j_{\text{infreq}})$ the event where j is the only infrequent type that arrives. On the event $\{C_j(\omega) = C\} \wedge \mathcal{F}^j_{\text{infreq}}$, note that (i) when (j,ℓ_j) arrive, we have only routed and matched frequent queries thus far, (ii) all type j queries have been routed to $C_j(\omega) = C$, and (iii) for all other query types $j' \neq j$ and $i \in [n]\setminus C$, we have

$$\Pr\left[\sum_{\ell=1}^{\ell_{j'}-1}\zeta_{j',\ell,i} = 1 \,\middle|\, C_j(\omega) = C, \mathcal{F}^j_{\text{infreq}}\right] = \sum_{\ell=1}^{\ell_{j'}-1}\Pr\left[\zeta_{j',\ell,i} = 1 \,\middle|\, \bar{\mathcal{E}}^{j'}_{\text{infreq}}\right]$$

$$
\begin{aligned}
&= \sum_{\ell=1}^{\ell_{j'}-1} \Pr\left[\zeta_{j',\ell,i}=1 \middle| D_{j'} \geq \ell\right] \Pr\left[D_{j'} \geq \ell \middle| D_{j'} \leq \ell_{j'}-1\right] \\
&\leq \sum_{\ell=1}^{\ell_{j'}-1} \Pr\left[\zeta_{j',\ell,i}=1 \middle| D_{j'} \geq \ell\right] \Pr\left[D_{j'} \geq \ell\right] \\
&= \Pr\left[\sum_{\ell=1}^{\ell_{j'}-1} \zeta_{j',\ell,i}=1\right] . \qquad (29)
\end{aligned}
$$

In particular, combining this inequality with Claim 5 for resource $i \in [n] \setminus C$ yields

$$
\Pr\left[\sum_{j' \neq j} \sum_{\ell=1}^{\ell_{j'}-1} \xi^{\uparrow}_{j',\ell,i}=1 \middle| C_j(\omega)=C, \mathcal{F}^j_{\text{infreq}}\right] \leq \frac{1+2\varepsilon}{2} , \qquad (30)
$$

where we remark that inequality (29), and the fact that routing decisions are independent across types $j' \neq j$, imply that the distribution of the set of queries of types $j' \neq j$ routed to resource i, conditional on $C^{\uparrow}_j(\omega)=C$ and $\mathcal{F}^j_{\text{infreq}}$, is stochastically dominated by the unconditional distribution of the set of queries of types $j' \neq j$ routed to the same i. Because Claim 5 considers a static threshold rule, this implies that the total match rate for resource i over all types $j' \neq j$ is at most $\frac{1+2\varepsilon}{2}$. Now, we observe that we can lower bound the rewards from infrequent items chosen by $\mathsf{ALG}^{\uparrow}_{\boldsymbol{E}}$ as follows:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} \sum_{\substack{\ell \geq 1: \\ (j,\ell) \in \hat{\mathcal{T}}_{\text{infreq}}}} r_{i,j} \xi^{\uparrow}_{j,\ell,i}(\omega)\right] \\
&\geq \sum_{j \in [m]} \sum_{\ell \geq \ell_j} \mathbb{E}\left[\sum_{C \subseteq [n]} \mathbb{I}\left[\mathcal{F}^j_{\text{infreq}}\right] \mathbb{I}[C^{\uparrow}_j(\omega)=C] \cdot \left(\sum_{u=1}^{U_j} \mathbb{I}[D_j \geq u+\ell_j-1] \cdot \mathbb{I}\left[\sum_{j' \neq j}\sum_{\ell=1}^{\ell_{j'}-1} \xi^{\uparrow}_{j',\ell,i}=0\right] r_{i^j_u}\right)\right] \\
&\geq \frac{(1-2\varepsilon)}{2} \cdot \mathbb{E}\left[\sum_{j \in [m]} \sum_{C \subseteq [n]} \sum_{u=1}^{U_j} \Pr\left[D_j \geq u+\ell_j-1, \mathcal{F}^j_{\text{infreq}} \middle| C^{\uparrow}_j(\omega)=C\right] \cdot r_{i^j_u}\right] \\
&\geq \frac{(1-4\varepsilon)}{2} \cdot \mathbb{E}\left[\sum_{j \in [m]} \sum_{u=1}^{U_j} \mathbb{I}\left[D_j \geq u+\ell_j-1\right] \cdot r_{i^j_u}\right] \\
&\geq \frac{(1-4\varepsilon)}{2} \cdot \mathbb{E}\left[\sum_{i \in [n]} \sum_{(j,\ell) \in \hat{\mathcal{T}}_{\text{infreq}}} r_{i,j} \delta^{\text{swap}}_{j,\ell,i}(\omega)\right] \\
&\geq \frac{(1-4\varepsilon)}{2} \cdot \left(\mathsf{LP}^{\mathsf{trunc}}(\mathbf{D}) - \mathsf{LP}^{\mathsf{trunc}}(\bar{\mathbf{D}})\right) ,
\end{aligned}
$$

where the first inequality proceeds from the fact that our algorithm on the event $\mathcal{F}^j_{\text{infreq}}$ selects the resources i^j_u by decreasing rewards, if they were not previously matched to a frequent query of type $j' \neq j$. The second inequality follows from inequality (30) and the independence of demand across types. The third inequality holds because $\Pr[D_j \geq u+\ell_j-1, \mathcal{F}^j_{\text{infreq}} | C^{\uparrow}_j(\omega)=C] \geq (1-2\varepsilon)\Pr[D_j \geq u+\ell_j-1 | C^{\uparrow}_j(\omega)=C]$ from inequality (26) and the independence of demand across types. The third inequality follows from (28). The last inequality is direct a consequence of Claim 6, by combining properties 1 and 2.

Appendix E: Extension to Network Revenue Management and Nonstationary Stochastic Horizon

Our result for CORREL extends to a Network Revenue Management setting where each matching option for an incoming query consumes multiple resources, and also allows a nonstationary stochastic horizon where the type distribution may vary over time.

In the nonstationary stochastic horizon setting, we are given a probability vector $(p_{t,j})_{j\in[m]}$ for each time $t\in[T]$ satisfying $\sum_{j\in[m]} p_{t,j} = 1$. The type of the new query in each time $t\in[T]$ is drawn independently of the history according to the probabilities $(p_{t,j})_{j\in[m]}$. Like before, the horizon length D is stochastic in that the sequence of queries may terminate after any time $t\in[T]$.

In the Network Revenue Management (NRM) extension, there are *products* $\rho\in\mathcal{P}$, where product ρ requires one unit each of a set of resources $A_\rho\subseteq[n]$ to produce. Type j yields a reward of $r_{\rho,j}$ whenever it is "matched" with product ρ, where $r_{\rho,j}=0$ indicates incompatibility between a type and product. We note that this is a more general version of NRM that includes "matching" decisions: that is, each arrival type j can be routed to a product ρ, generating a reward $r_{\rho,j}$. The basic version of NRM is captured by setting $\mathcal{P}=\{1,\dots,m\}$ and having $r_{\rho,j}=0$ unless $\rho=j$—in that simplified setting, there is a single reward per product/customer type. On the other hand, the online matching problem (without NRM) is captured by setting $\mathcal{P}=\{1,\dots,n\}$ and $A_\rho=\{\rho\}$ for all products/resources $\rho\in[n]$. (In terms of whether our INDEP result extends to NRM: we believe our Lossless Rounding naturally extends, but we are unsure how to efficiently write/solve the analogue of our $\mathsf{LP}^{\mathsf{trunc}}$ when resource constraints are coupled, so we leave this as an open problem.)

DEFINITION 3. For any instance $\mathcal{I}$ of NRM with nonstationary stochastic horizons, we define $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ as the optimal objective value of the following LP:

$$\begin{aligned}
\max \quad & \sum_{j=1}^{m}\sum_{\rho\in\mathcal{P}} r_{\rho,j}\sum_{t=1}^{T}\Pr[D\ge t]\cdot y_{\rho,j}^{t} && \\
\text{s.t.} \quad & \sum_{j=1}^{m}\sum_{\rho\in\mathcal{P}:i\in A_\rho}\sum_{t=1}^{T} y_{\rho,j}^{t}\le k_i && \forall i\in[n]\\
& \sum_{\rho\in\mathcal{P}} y_{\rho,j}^{t}\le p_{t,j} && \forall j\in[m],\forall t\in[T]\\
& y_{\rho,j}^{t}\ge 0 && \forall \rho\in\mathcal{P},\forall j\in[m],\forall t\in[T]\ .
\end{aligned}$$

Decision variable $y_{\rho,j}^{t}$ represents the probability of matching a query of type j to product ρ at time t, conditional on $D\ge t$. Using the same proof as in Appendix C.5 that conditions on $D=T$, we can see that $\mathsf{ALG}(\mathcal{I})\le\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ for all online algorithms and hence $\mathsf{OPT}(\mathcal{I})\le\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$.

Our approximate online algorithm is then given in Algorithm 4, where the comments highlight the changes from Algorithm 3.

We now require every resource in A_ρ to accept in order to match a product ρ, where each resource i runs its own OCRS with a starting inventory of k_i units (ignoring the fact that the NRM problem "couples" the resources). We recall that if $\mathsf{Acc}_{i,t}(\mathsf{inv}[i])=1$ then it is guaranteed for resource i to have remaining inventory.

Algorithm 4 Algorithm for NRM matching with nonstationary stochastic horizon

Solve $\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})$ from Definition 3, letting $(y^t_{\rho,j})_{t\in[T],\rho\in\mathcal{P},j\in[m]}$ denote an optimal solution

$\mathsf{inv}[i] = k_i$ for all $i \in [n]$

for $t = 1, \ldots, D$ **do**

Let j denote type of query t ▷ nonstationary probability vector $(p_{t,j})_{j\in[m]}$

Choose ρ independently from probability vector $\left(\frac{y^t_{\rho,j}}{p_{t,j}}\right)_{\rho\in\mathcal{P}}$ ▷ ρ is a product (set ρ to $\perp$ with probability $1 - \sum_{\rho\in\mathcal{P}} \frac{y^t_{\rho,j}}{p_{t,j}}$)

if $\rho \neq \perp$ and $\mathsf{Acc}_{i,t}(\mathsf{inv}[i]) = 1$ for all $i \in A_\rho$ **then** ▷ every resource in A_ρ must accept

Match the query to product ρ, collecting reward $r_{\rho,j}$

$\mathsf{inv}[i] = \mathsf{inv}[i] - 1$ for all $i \in A_\rho$

end if

end for

Similarly to our original analysis of Section C, we consider an alternative world where time t always runs until the maximum T but we only account for the rewards collected in times $t \le D$:

$$\begin{aligned}
\mathsf{ALG}(\mathcal{I}) &= \sum_{t=1}^{T}\sum_{j=1}^{m} p_{t,j} \sum_{\rho\in\mathcal{P}} \frac{y^t_{\rho,j}}{p_{t,j}} \Pr\left[\bigcap_{i\in A_\rho} (\mathsf{Acc}_{i,t}(\mathsf{inv}[i]) = 1)\right] r_{\rho,j} \Pr[D \ge t] \\
&= \sum_{t=1}^{T}\sum_{j=1}^{m}\sum_{\rho\in\mathcal{P}} y^t_{\rho,j} \left(1 - \Pr\left[\bigcup_{i\in A_\rho} (\mathsf{Acc}_{i,t}(\mathsf{inv}[i]) = 0)\right]\right) r_{\rho,j} \Pr[D \ge t] \\
&\ge \sum_{t=1}^{T}\sum_{j=1}^{m}\sum_{\rho\in\mathcal{P}} y^t_{\rho,j} \left(1 - \sum_{i\in A_\rho} (1 - \Pr[(\mathsf{Acc}_{i,t}(\mathsf{inv}[i]) = 1)])\right) r_{\rho,j} \Pr[D \ge t] \\
&\ge \sum_{t=1}^{T}\sum_{j=1}^{m}\sum_{\rho\in\mathcal{P}} y^t_{\rho,j} \left(1 - \sum_{i\in A_\rho} (1 - \gamma^*_{k_i})\right) r_{\rho,j} \Pr[D \ge t] \\
&\ge (1 - L(1 - \gamma^*_k))\mathsf{LP}^{\mathsf{cond}}(\mathcal{I})
\end{aligned}$$

where the first inequality applies the union bound, the second inequality follows from the OCRS guarantee, and the final guarantee assumes that $|A_\rho| \le L$ for all ρ, and $k_i \ge k$ for all i (recall that the tight OCRS guarantee γ^*_k is increasing in k). This results in the following theorem.

THEOREM 6. *For the nonstationary stochastic horizon and NRM extensions, Algorithm 4 is a polynomial-time online algorithm that is $(1 - L(1 - \gamma^*_k))$-approximate for every instance in which all products $\rho \in \mathcal{P}$ require at most $|A_\rho| \le L$ resources and all resources $i \in [m]$ start with inventory at least $k_i \ge k$.*

We note that the preceding argument is based on the recent work of Amil et al. (2022), extending their performance guarantee to the stochastic horizon setting. Theorem 6 can also be seen as improving the guarantee from Bai et al. (2022) in two ways: extending NRM to have "matching" decisions, and improving

the convergence rate to 1 when L is fixed and k tends to ∞. To see this, note that $\gamma_k^* \geq 1 - 1/\sqrt{k+3}$ (Jiang et al. 2022, Alaei et al. 2012a), and thus, our algorithm's approximation ratio is at least

$$1 - \frac{L}{\sqrt{k+3}} = 1 - O_L\left(\sqrt{\frac{1}{k}}\right)$$

whereas Bai et al. (2022) establish an approximation ratio that is $1 - O_L(\sqrt{\frac{\log k}{k}})$.

Appendix F: Concrete Examples of our Algorithmic Advancements

We construct simple examples illustrating why our algorithms may outperform existing ones under the Indep and Correl models. Although these are toy examples, they may reasonably depict practical tradeoffs in real-world scenarios. In particular, Example 2 represents a resource allocation setting where a central planner would like to maximize welfare, but the tradeoff is that the most valuable resources (which generate the largest rewards when allocated) are also the most flexible and perhaps should be saved for the most picky customer types. Example 3 is a typical service operations setting where two demand types each have a dedicated server and also share a common server. Finally, Example 4 is a revenue management setting where k units of a single item can be dynamically priced, except that there could be a large common shock to the aggregate demand, i.e., the item is either a runaway hit or a total flop.

Example 2 (Differences between $\mathsf{LP}^{\mathsf{trunc}}$ and fluid LP). Consider a simple example with two resources and two demand types. Resource 1 is the superior resource, compatible with both demand types and yielding a high reward when matched. Resource 2 is the inferior resource, compatible with only demand type 2 (the less picky type) and yielding a low reward when matched. Suppose that both resources start with k units and the expected demand of each type is exactly k. The fluid LP, optimizing for a world with no demand variability, would suggest dedicating resource 1 to serving type 1 and resource 2 to serving type 2, perfectly matching supply and demand. However, this dedicated service is undesirable in the true stochastic environment with demand variability—it could leave the most valuable resources unutilized with a large probability. By contrast, our $\mathsf{LP}^{\mathsf{trunc}}$ allocates a portion of resource 1 to demand type 2 to increase the probability that it is utilized; the portion size depends on the level of demand variance and the ratio of high to low reward.

To formalize this, let $n = 2$, $m = 2$, $r_{1,1} = r_{1,2} = r_H$, and $r_{2,2} = r_L, r_{2,1} = 0$. Here we let r_H, r_L denote the High, Low rewards with $r_H \geq r_L$, and resource 2 is the inferior one, which cannot serve the picky type 1. The starting inventory is set as $k_1 = k_2 = k$ for some positive integer k, and let D_1, D_2 both be independent Poisson random variables with mean k, denoted by $\text{Pois}(k)$. Under the dedicated service policy described by the optimal fluid LP solution, total expected reward is

$$r_H \mathbb{E}\left[\min\{\text{Pois}(k), k\}\right] + r_L \mathbb{E}\left[\min\{\text{Pois}(k), k\}\right] = (r_H + r_L)k\left(1 - e^{-k}\frac{k^k}{k!}\right). \tag{31}$$

However, this is quite suboptimal when r_H is much larger than r_L, because the expected fraction of the superior resource that goes untilized is $e^{-k}k^k/k! = \Theta((2\pi k)^{-\frac{1}{2}})$, which decays slowly as a function of k. Strikingly, an "extreme policy" that always allocates the superior resource has total expected reward $r_H \mathbb{E}[\min\{\text{Pois}(2k), k\}]$, which can be higher than the expression in (31) when the ratio r_H/r_L is sufficiently

large. In contrast $\mathsf{LP}^{\mathsf{trunc}}$ balances between the two extremes of maximizing utilization, i.e., maximizing expected demand served via dedicated service versus maximizing the expected amount of resource 1 allocated. Depending on the ratio r_H/r_L and the exact demand distributions, our $\mathsf{LP}^{\mathsf{trunc}}$ prescribes a solution that lies in-between these two extremes, partially using resource 1 to serve type 2.

It is not easy to quantify the exact benefit of $\mathsf{LP}^{\mathsf{trunc}}$ in the Poisson setting as a closed-form expression. To exhibit concrete numbers, we consider a modification to the preceding example where $k=1$, D_1 is equally likely to be 0 or 2, and D_2 is deterministically 1. The fluid LP still suggests dedicated service, with an expected total reward of $r_H/2+r_L$. In contrast, if type 1 queries arrive before the type 2 query, our truncated LP yields the exact optimal solution $x_{1,1}=x_{1,2}=x_{2,2}=1/2, x_{2,1}=0$. Now, if the type 2 query arrives before type 1 queries, the expected reward would still be $(3/4)r_H+r_L/2$, exceeding $r_H/2+r_L$ as long as $r_H/r_L>2$.
□

We remark that in Example 2, the fluid LP's solution was undesirable even though the Poisson demand distribution was "low variance" in that its variance is no greater than its mean.

Example 3 (Lossless Rounding outperforms Independent Rounding). First, we explain the shortcomings of Independent Rounding and its "In-stock" improvement that was tested in Section 6. Suppose for a type j that $D_j \in \{1,2\}$ with $\Pr[D_j=2]=\rho$ and $\Pr[D_j=1]=1-\rho$. Consider a fractional solution where $x_{j,j}=1$ and $x_{0,j}=\rho$, with j and 0 being different resources (note that this solution satisfies constraints (7) in our $\mathsf{LP}^{\mathsf{trunc}}$). Independent Rounding routes each arrival of type j independently to resources j and 0 with probabilities $1/(1+\rho)$ and $\rho/(1+\rho)$ respectively. This preserves the marginals in that the expected numbers routed to resources j and 0 are 1 and ρ, which equal $x_{j,j}$ and $x_{0,j}$, respectively; however, the downside is that multiple arrivals may be routed to the same resource and that resource may only start with one unit of inventory. In-stock Independent Rounding routes the first arrival of type j to resources j and 0 with probabilities $1/(1+\rho)$ and $\rho/(1+\rho)$ respectively, but differs by always routing the second arrival to the other resource. This guarantees to route to different resources; however, marginals are not preserved—it can be checked that the expected numbers routed to resources j and 0 are $(1+\rho^2)/(1+\rho)$ and $2\rho/(1+\rho)$, which do not equal 1 and ρ respectively unless[11] $\rho\in\{0,1\}$. Our Lossless Rounding satisfies both the criteria of preserving the marginals and routing to different resources, which on this example means routing the first arrival of type j to resource j, and the second to resource 0.

To see why this can improve the overall performance, consider an example where resources $i=0,1,2$ each have starting inventory $k_i=1$. There are $m=2$ demand types with $r_{1,1}=r_{2,2}=r_{0,1}=r_{0,2}=1$, and all other rewards are 0. Demands D_1 and D_2 are each independently equal to 1 or 2 with probabilities $1-\rho$ and ρ, respectively. An optimal solution to $\mathsf{LP}^{\mathsf{trunc}}$ sets $x_{1,1}=x_{2,2}=1, x_{0,1}=x_{0,2}=\rho$, and all unspecified $x_{i,j}$'s are equal to 0. Regardless of arrival order, our Lossless Rounding always matches resources 1,2 and matches resource 0 with probability $1-(1-\rho)^2$, for a total expected reward of

$$2+2\rho-\rho^2. \tag{32}$$

[11] The condition $\rho\in\{0,1\}$ means that D_j is deterministic. Even in this case, it is possible to construct examples where In-stock Independent Rounding does not satisfy both criteria. Indeed, suppose $\Pr[D_j=2]=1$ and there are 3 resources with $x_{1,j}=1, x_{2,j}=x_{3,j}=1/2$. In-stock Independent Rounding routes to resource set $\{2,3\}$ with probability $\frac{1/2+1/2}{1+1/2+1/2}\cdot\frac{1/2}{1+1/2}=1/6$, which means that resource 1 only gets routed a query with probability 5/6.

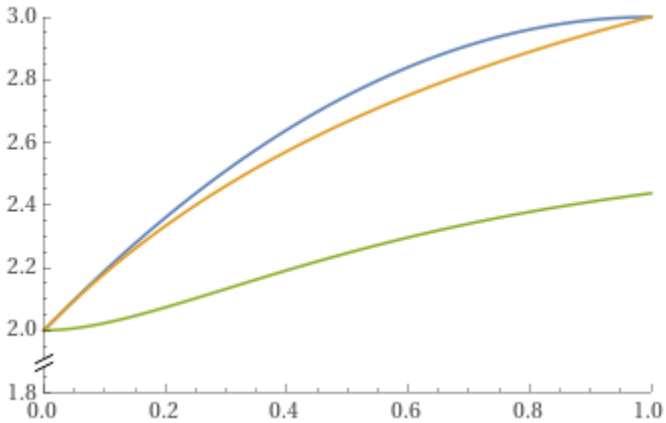
Independent Rounding would match each resource $1,2$ with probability $1-\frac{\rho}{1+\rho}(1-\rho\frac{1}{1+\rho}) = \frac{1+\rho+\rho^2}{(1+\rho)^2}$ and resource 0 with probability $1-(\frac{1}{1+\rho})^2(1-\rho\frac{\rho}{1+\rho})^2 = 1-\frac{(1+\rho-\rho^2)^2}{(1+\rho)^4}$, for a total expected reward of

$$2\frac{1+\rho+\rho^2}{(1+\rho)^2}+1-\frac{(1+\rho-\rho^2)^2}{(1+\rho)^4}. \tag{33}$$

In-stock Independent Rounding has a more nuanced state trajectory that depends on the arrival order. Whichever type j arrives first, it will be routed to resource 0 with probability $\rho/(1+\rho)$, conditional on which the total expected reward is $2+\rho$ (resources 0 and $2-j$ get matched; resource j gets matched if there is another arrival of type j). On the other hand, conditional on the first arrival not being routed to resource 0, the best-case arrival order is that the same type arrives again. If it does (happening with probability ρ), then the total reward is 3; otherwise, the conditional total expected reward is again $2+\rho$. Putting it together, the unconditional total expected reward of In-stock Independent Rounding is

$$\begin{aligned}\frac{\rho}{1+\rho}(2+\rho)+\frac{1}{1+\rho}\rho(3)+\frac{1}{1+\rho}(1-\rho)(2+\rho) &= \frac{2+\rho}{1+\rho}+\frac{3\rho}{1+\rho}\\ &= 2+2\rho-\frac{2\rho^2}{1+\rho}.\end{aligned} \tag{34}$$

We compare these expressions in the following plot, where the best performance (highest curve) is attained by Lossless Rounding (eq. (32)), the second-best performance is attained by In-stock Independent Rounding (eq. (34)), and the worst performance is attained by Independent Rounding (eq. (33)).



Our Lossless Rounding outperforms In-stock Independent Rounding even under its best-case arrival order, and the difference in performance is most pronounced for medium values of ρ, which represent the highest variance of the demand distribution. □

Example 4 (Differences between $\mathsf{LP}^{\mathsf{cond}}$ and fluid LP). Support there is a single resource with k units, where k is a large even positive integer. We will omit the resource index i. Consider our Correl model where the total demand D is equally likely to be $k/2$ or $3k/2$. There are two demand types indexed by $j=L$ (Low) and $j=H$ (High), with $r_L<r_H$ and $p_L=1/3, p_H=2/3$. Similar conclusions could be drawn for an arbitrary price curve, but this simple price curve is sufficiently illustrative.

We first explain our algorithm's decisions. Our $\mathsf{LP}^{\mathsf{cond}}$ can be expressed as follows, after defining aggregate variables $y_j^{\mathsf{pre}} := \sum_{t=1}^{k/2} y_j^t$ and $y_j^{\mathsf{post}} := \sum_{t=k/2+1}^{3k/2} y_j^t$ for $j\in\{L,H\}$:

$$\max\ r_L y_L^{\mathsf{pre}} + r_H y_H^{\mathsf{pre}} + \frac{r_L}{2} y_L^{\mathsf{post}} + \frac{r_H}{2} y_H^{\mathsf{post}}$$

$$\begin{aligned} \text{s.t. } & y_L^{\mathsf{pre}} + y_L^{\mathsf{post}} + y_H^{\mathsf{pre}} + y_H^{\mathsf{post}} \le k \\ & 0 \le y_L^{\mathsf{pre}} \le k/6 \\ & 0 \le y_H^{\mathsf{pre}} \le k/3 \\ & 0 \le y_L^{\mathsf{post}} \le k/3 \\ & 0 \le y_H^{\mathsf{post}} \le 2k/3 \ . \end{aligned}$$

The optimal solution always sets y_H^{pre} to its upper bound of $k/3$, because this variable has the largest objective coefficient of r_H. The optimal solution would then fill variables $y_L^{\mathsf{pre}}, y_H^{\mathsf{post}}$ to their upper bounds, with the prioritization depending on whether $r_L \ge r_H/2$. Indeed, if $r_L \ge r_H/2$, then it is optimal to set $y_L^{\mathsf{pre}} = k/6, y_H^{\mathsf{post}} = k/2, y_L^{\mathsf{post}} = 0$ (because $k/3 + k/6 + k/2 = k$), in which case the algorithm would accept both the High and Low types at the beginning, and then only accept the High type conditional on surviving past time $k/2$ (i.e, conditional on D realizing to $3k/2$). On the other hand, if $r_L < r_H/2$, then it is optimal to set $y_L^{\mathsf{pre}} = 0, y_H^{\mathsf{post}} = 2k/3, y_L^{\mathsf{post}} = 0$, in which case the algorithm would only accept High types at the beginning, so that a maximum amount of resource is preserved in case D realizes to $3k/2$.

The fluid LP, by contrast, sets $y_L^{\mathsf{pre}} = k/6$ (i.e., accepting all Low types at the beginning) regardless of whether $r_L \ge r_H/2$, because the expected demand is k which is no greater than the starting inventory. If $r_L < r_H/2$, then the fluid LP foregoes $(r_H/2 - r_L)k/6$ in expected rewards[12] compared to using our $\mathsf{LP}^{\mathsf{cond}}$. It is worth noting that this logic holds even for an "intelligent re-solving" heuristic, which continuously reoptimizes the fluid LP based on the remaining inventory and conditional expected demand; in that context, only High types are accepted after the $k/2$-th arriving query. The fluid LP foregoes revenue because it is agnostic to the variance of D; it fails to compare the ratio r_H/r_L of high to low reward to the probability $1/2$ that the demand is large $D = 3k/2$. In fact, there is a newsvendor-like tradeoff in choosing the service levels in the first stage and second stage of the demand realization. Although the distribution of D in this example is quite stylized to ease calculations, we believe that the intuition can be extrapolated to more general demand distributions. $\square$

Appendix G: Additional Proofs from Section 4

G.1. Proof of Proposition 1

We first present construction (i). Consider a family of instances parametrized by $\varepsilon \in (0,1)$. Each instance comprises a single resource $i = 1$ with starting inventory $k_i = 1$ and a single query type $j = 1$ with the normalized reward $r_{1,1} = 1$. The demand random variable D_1 takes two values $\{0, \lceil \frac{1}{\varepsilon} \rceil\}$ with probabilities $\Pr[D_1 = 0] = 1 - \varepsilon$ and $\Pr[D_1 = \frac{1}{\varepsilon}\rceil] = \varepsilon$. Because $\mathbb{E}[D_1] = 1$, it is clear that $\mathsf{LP}(\mathcal{I}) = 1$. However, the single resource can be matched to at most one query, which only occurs with probability $\Pr[D_1 > 0] = \Pr[D_1 = \frac{1}{\varepsilon}\rceil] = \varepsilon$. We have just shown that $\frac{\mathsf{OFF}(\mathcal{I})}{\mathsf{LP}(\mathcal{I})} = \varepsilon$, which proves part (i) of Proposition 1.

We now present our construction for part (ii). Fix a large n, and let $k_i = 1$ for all $i \in [n]$. Let $m = 1$, and let $r_{i,1} = 0$ unless $i = 1$, in which case $r_{1,1} = 1$. Let D_1 take the value n with probability $\frac{1}{n}$, and take the value 0

[12] As a rough approximation, we assume that a fraction exactly 1/3 or 2/3 of the arrivals has types L and H, respectively. There could be $O(\sqrt{k})$-deviations from this quantity, which are of second order compared to the fluid LP's loss of $(r_H/2 - r_L)k/6$ when k is large.

with the residual probability $1-\frac{1}{n}$. Note that D_1 is indeed no greater than $\sum_{i=1}^{n} k_i = n$ with probability 1. It is feasible in the fluid relaxation LP to set $x_{1,1}=1$, yielding an objective value 1. Meanwhile, any algorithm can match query type 1 to resource type 1 with probability at most $\frac{1}{n}$ (when $D_1 = n$). Therefore, we have shown that $\frac{\mathsf{OFF}(\mathcal{I})}{\mathsf{LP}(\mathcal{I})} = \frac{1}{n}$, where n can be arbitrarily large, thereby completing the proof of Proposition 1.

G.2. Proof of Lemma 1

To show that $\mathsf{OFF}(\mathcal{I}) \le \mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$, we represent the offline-optimum as the output of a certain exponentially sized linear program. The offline benchmark solves a max-weight matching problem with respect to any specific realization $\mathbf{d}$ of the demand $\mathbf{D}$. Letting $\bar{\mathbf{d}} = \{\mathbf{d} : \Pr[\mathbf{D}=\mathbf{d}>0]\}$ be the support of $\mathbf{D}$, the offline benchmark can be formulated as:

$$\begin{aligned}
\mathsf{OFF}(\mathcal{I}) = \quad \max \quad & \sum_{\mathbf{d}\in\bar{\mathbf{d}}} \Pr[\mathbf{D}=\mathbf{d}] \cdot \left(\sum_{i=1}^{n}\sum_{j=1}^{m} r_{i,j} x_{i,j}^{\mathbf{d}} \right) \\
\text{s.t.} \quad & \sum_{j=1}^{m} x_{i,j}^{\mathbf{d}} \le k_i && \forall i \in [n], \forall \mathbf{d}\in\bar{\mathbf{d}} \qquad (35)\\
& \sum_{i=1}^{n} x_{i,j}^{\mathbf{d}} \le \mathbf{d}_j && \forall j \in [m], \forall \mathbf{d}\in\bar{\mathbf{d}} \qquad (36)\\
& x_{i,j}^{\mathbf{d}} \ge 0 && \forall i \in [n], \forall j \in [m], \forall \mathbf{d}\in\bar{\mathbf{d}} \ .
\end{aligned}$$

Fix a feasible solution $(x_{i,j}^{\mathbf{d}})_{i\in[n],j\in[m],\mathbf{d}\in\bar{\mathbf{d}}}$ of the above LP. For all $\mathbf{d}\in\bar{\mathbf{d}}$, $j\in[m]$, and $S\subseteq[n]$, we have $\sum_{i\in S} x_{i,j}^{\mathbf{d}} \le \sum_{i=1}^{n} x_{i,j}^{\mathbf{d}} \le \mathbf{d}_j$ based on constraint (36). By summing inequalities (35) over all $i\in S$, we infer that $\sum_{i\in S} x_{i,j}^{\mathbf{d}} \le \sum_{i\in S}\sum_{j=1}^{m} x_{i,j}^{\mathbf{d}} \le \sum_{i\in S} k_i$. By combining these inequalities, we infer that any solution of the offline LP must satisfy $\sum_{i\in S} x_{i,j}^{\mathbf{d}} \le \min\{\mathbf{d}_j, \sum_{i\in S} k_i\}$. Now, we consider the vector $(x_{i,j})_{i\in[n],j\in[m]}$ with $x_{i,j} = \sum_{\mathbf{d}\in\bar{\mathbf{d}}} \Pr[\mathbf{D}=\mathbf{d}] \cdot x_{i,j}^{\mathbf{d}}$, obtained as the weighted sum of the offline assignment variables. Based on the previous observation, for each $S\subseteq[n]$ and $j\in[m]$, we have

$$\sum_{i\in S} x_{i,j} = \sum_{\mathbf{d}\in\bar{\mathbf{d}}} \Pr[\mathbf{D}=\mathbf{d}] \cdot \left(\sum_{i\in S} x_{i,j}^{\mathbf{d}} \right) \le \sum_{\mathbf{d}\in\bar{\mathbf{d}}} \Pr[\mathbf{D}=\mathbf{d}] \cdot \min\left\{ \mathbf{d}_j, \sum_{i\in S} k_i \right\} = \mathbb{E}\left[\min\left\{ D_j, \sum_{i\in S} k_i \right\} \right] \ ,$$

which indicates that $(x_{i,j})_{i\in[n],j\in[m]}$ satisfies constraint (7) of $\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$. It is straightforward to verify that all other constraints of $\mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$ are also met by $(x_{i,j})_{i\in[n],j\in[m]}$. By exploiting this mapping, it follows that $\mathsf{OFF}(\mathcal{I}) \le \mathsf{LP}^{\mathsf{trunc}}(\mathcal{I})$.

G.3. Proof of Claim 1

We seek to establish that for any $j\in[m]$, if we re-index i so that $\frac{x_{1,j}}{k_1} \ge \cdots \ge \frac{x_{n,j}}{k_n}$, then the exponential family of inequalities (9) for type j is implied by the following linearly many inequalities:

$$x_{1,j} + \cdots + x_{i,j} \le \sum_{\ell=1}^{k_1+\cdots+k_i} \Pr[D_j \ge \ell] \qquad \forall i \in [n] \ . \qquad (37)$$

In fact, we show that (37) implies a family of constraints which is even more general:

$$\sum_{i=1}^{n} S_i \frac{x_{i,j}}{k_i} \le \sum_{\ell=1}^{S_1+\cdots+S_n} \Pr[D_j \ge \ell] \qquad \forall S_1 \in [k_i]\cup\{0\}, \ldots, S_n \in [k_n]\cup\{0\} \ . \qquad (38)$$

To show that (37) implies (38), take any inequality in (38) and let ℓ' denote $S_1+\cdots+S_n$, which we assume is non-zero. We must show that the LHS of (38) is no greater than $\sum_{\ell=1}^{\ell'} \Pr[D_j \ge \ell]$. Notice that the LHS

of (38) can be interpreted as selecting S_i units of every resource $i \in [n]$, and accumulating $x_{i,j}/k_i$ for each unit of i selected. From this interpretation, we can see that subject to the constraint $S_1 + \cdots + S_n = \ell'$, the LHS of (38) is maximized by setting $S_i = k_i \ \forall i < i'$, $S_{i'} = \ell' - \sum_{i<i'} k_i$, $S_i = 0 \ \forall i > i'$, for some unique index i' in $[n]$ such that this definition of $S_{i'}$ lies in $[k_{i'}]$ (this is because the indices i are sorted in decreasing order of $x_{i,j}/k_i$). Adding $1 - S_{i'}/k_{i'}$ times constraint (37), where we set $i = i' - 1$ (with the understanding that this constraint is $0 \le 0$ if $i' = 1$), to $S_{i'}/k_{i'}$ times constraint (37), where we set $i = i'$, we obtain

$$\begin{aligned}\sum_{i<i'} x_{i,j} + \frac{S_{i'}}{k_{i'}} x_{i',j} &\le \sum_{\ell=1}^{k_1+\cdots+k_{i'-1}} \Pr[D_j \ge \ell] + \frac{S_{i'}}{k_{i'}} \sum_{\ell=k_1+\cdots+k_{i'-1}+1}^{k_1+\cdots+k_{i'}} \Pr[D_j \ge \ell] \\ &\le \sum_{\ell=1}^{k_1+\cdots+k_{i'-1}} \Pr[D_j \ge \ell] + \sum_{\ell=k_1+\cdots+k_{i'-1}+1}^{k_1+\cdots+k_{i'-1}+S_{i'}} \Pr[D_j \ge \ell]\end{aligned}$$

where the second inequality holds because $\Pr[D_j \ge \ell]$ is decreasing in ℓ. This shows that even by maximizing the LHS of (38), we cannot exceed its RHS for this value of ℓ', completing the proof that (37) implies (38) which in turn implies (9).

The other direction of Claim 1 is immediate because (9) is a superset of the constraints in (37). This completes the proof of Claim 1.

G.4. Proof of Proposition 2

Let $n = 1$ and fix any starting inventory k_1, which can be arbitrarily large. There are $m = 2$ query types, with $r_{1,1} = 1$ and $r_{1,2} = 1/\varepsilon$, for some small $\varepsilon > 0$. The demand vector $\mathbf{D}$ realizes to (k_1, k_1) with probability ε, and $(k_1, 0)$ with the residual probability $1 - \varepsilon$. The arrival order is such that all queries of type 2 arrive after all queries of type 1.

The offline optimum collects reward $\frac{k_1}{\varepsilon}$ when $D_2 = k_1$, and k_1 when $D_2 = 0$, which implies in expectation that

$$\mathsf{OFF}(\mathcal{I}) = \varepsilon \cdot \frac{k_1}{\varepsilon} + (1-\varepsilon) \cdot k_1 = (2-\varepsilon) k_1 \ .$$

Meanwhile, any online algorithm does not know whether D_2 will be k_1 or 0 during the first k_1 arrivals, corresponding to all queries of type 1. Suppose it accepts k queries of type 1, for some $k \le k_1$. Then, it will collect expected rewards $k + \varepsilon \cdot (k_1 - k) \cdot \frac{1}{\varepsilon} = k_1$. Therefore, $\mathsf{ALG}(\mathcal{I}) \le k_1 = \frac{1}{2-\varepsilon} \cdot \mathsf{OPT}(\mathcal{I})$, and taking the limit $\varepsilon \to 0$ completes the proof.

Appendix H: Supplement to Simulation Section 6

Offline LP: we employ variables $(x_{i,j})_{i \in [n], j \in [m]}$ from an optimal solution to the following LP:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} r_{i,j} x_{i,j} & & \\
\text{s.t.} \quad & x_{i,j} = \frac{1}{M} \sum_{s=1}^{M} x_{i,j}^s & & \forall i \in [n], \forall j \in [m] \\
& \sum_{j=1}^{m} x_{i,j}^s \le k_i & & \forall i \in [n], \forall s \in [M] \\
& \sum_{i=1}^{n} x_{i,j}^s \le D_j^s & & \forall j \in [m], \forall s \in [M] \\
& x_{i,j}^s \ge 0 & & \forall i \in [n], \forall j \in [m], \forall s \in [M] \ .
\end{aligned}$$

In the offline LP, each s indexes a demand scenario $(D_j^s)_{j\in[m]}$, and $(x_{i,j}^s)_{i\in[n],j\in[m]}$ is an optimal offline matching under that scenario. The employed variable $x_{i,j}$ is the average value of $x_{i,j}^s$ across scenarios s, and the optimal objective value coincides with the expected offline value that we earlier denoted using $\mathsf{OFF}(\mathcal{I})$ (assuming the M scenarios fully capture the distribution of $(D_j)_{j\in[m]}$). Although the Fluid LP is convenient for theoretical analysis, it has been observed that the Offline LP usually improves empirical performance (Talluri and Van Ryzin 1999, DeValve et al. 2021).

The feasible region of the Offline LP lies within that of the Truncated LP. Indeed, constraints (7) in $\mathsf{LP}^{\mathsf{trunc}}$ are satisfied if the M demand scenarios fully capture the distribution of $(D_j)_{j\in[m]}$, because the Offline LP's solution satisfies $\sum_{i\in S} x_{i,j}^s \le \min\{D_j^s, \sum_{i\in S} k_i\}$ for every scenario s. Therefore, we can use Lossless Rounding also on the Offline LP's solution.

Defining the LP's for different instances: For INDEP synthetic instances, it is clear how to define the Fluid and Truncated LP's, and we omit the Conditional LP. For CORREL synthetic instances, it is clear how to define the Fluid and Conditional LP's, and we omit the Truncated LP. In the Offline LP, we always sample $M = 100$ times for the synthetic distributions, which appears sufficient to capture the true distributions.

For real-data instances, to define the Fluid and Truncated LP's, we use the true distribution of (up to 12) monthly demands from a location j (for the SKU) to evaluate the expectations involving D_j. For the Conditional LP, we evaluate probabilities involving $D = \sum_{j=1}^m D_j$ in the same way, and set each p_j to be the fraction of all orders across months (for the SKU) to come from location j. For the Offline LP, each demand scenario $(D_j^s)_{j\in[m]}$ represents the actual demands over a month $s \in [M]$ (where $M \le 12$). We remark that in these ways of defining the LP's from real data, the Fluid and Truncated LP's are agnostic to the correlation of D_j across j, the Conditional LP is agnostic to how a given D_j may vary across months if it is not proportional to D, while the Offline LP is given the most information.

Defining q_i for the IR variants: We have $q_i = x_{i,j}/\mathbb{E}[D_j]$ for the Fluid LP, $q_i = x_{i,j}/\mathbb{E}[\min\{D_j, \sum_{i'=1}^n k_{i'}\}]$ for the Truncated LP, $q_i = y_{i,j}^t/p_j$ for the Conditional LP, and $q_i = x_{i,j}/(\frac{1}{M}\sum_{s=1}^M D_j^s)$ for the Offline LP. (Note that q_i depends on the time t for the Conditional LP.)

Motivation for testing In-stock IR: This rounding method is intuitive and is also known as "boosting" (Ma et al. 2023) or "sampling without replacement" (Huang et al. 2024), with both papers demonstrating its strong empirical performance.

Appendix I: De-randomization Methods

INDEP *model: De-randomizing Algorithms 1 and 2.* De-randomizing Algorithm 1 is non-trivial because an adversary selects the arrival order, and hence simulation into the future is no longer possible. Nonetheless, technically there should exist a deterministic algorithm that is 1/2-competitive because our performance guarantee holds even against an almighty adversary that knows the realizations of the random algorithm's bits. Therefore, letting ALG denote the *realized deterministic* algorithm, we have

$$\begin{aligned} & & \mathbb{E}_{\mathbf{D},\mathsf{ALG}}[\inf_\sigma \mathsf{ALG}_{\mathcal{I}}(\mathbf{D},\sigma)] &\ge \mathbb{E}_{\mathbf{D}}[\mathsf{OFF}(\mathbf{D})]/2 \\ &\implies & \mathbb{E}_{\mathsf{ALG}}[\mathbb{E}_{\mathbf{D}}[\inf_\sigma \mathsf{ALG}(\mathbf{D},\sigma)]] &\ge \mathbb{E}_{\mathbf{D}}[\mathsf{OFF}(\mathbf{D})]/2 \\ \exists \mathsf{ALG} \text{ (deterministic) s.t.} & & \mathbb{E}_{\mathbf{D}}[\inf_\sigma \mathsf{ALG}(\mathbf{D},\sigma)] &\ge \mathbb{E}_{\mathbf{D}}[\mathsf{OFF}(\mathbf{D})]/2\ . \end{aligned}$$

That said, finding the deterministic algorithm that is 1/2-competitive could be algorithmically intractable. Fortunately, the use of randomness in Algorithm 2 is very explicit. The lossless rounding uses n random bits per resource, so nm random bits in total. Therefore, applying the method of conditional expectations (Alon and Spencer 2016, Chap. 16)—fixing the value of each random bit to 1 or 0 iteratively—can achieve a competitive ratio if $\frac{1}{2} - O(\varepsilon)$. To do so, we must be able to simulate the expected total rewards for a *worst-case ordering* on each realization of the demand $\mathbf{D}$ and each draw of the random bits in our lossless rounding. Now, this is feasible because our prophet inequality algorithm and lossless rounding decouple how the adversary chooses "permutations" across resources: we let the almighty adversary choose a distinct permutation for each resource after we reveal which queries have been routed to it. Note that these permutations could be inconsistent across types. Regardless, we independently sample a pseudo-polynomial number of simulation draws to estimate the expected total rewards, while losing only $O(\frac{\varepsilon}{nm})$ in the approximation factor in each of the nm iterations.

CORREL *model: De-randomizing Algorithm 3.* De-randomizing Algorithm 3 for CORREL $\cap$ RAND is relatively easier since there is no adversarial element. At each time t, both the resource i that the query is routed to, and whether the resource accepts the query, can be determined by a random algorithmic decision. In principle, de-randomization is possible—there are only $n+1$ ultimate decisions (matching to one of n resources, or rejection), and the future from each one can be *simulated* (still using the same *randomized* algorithm for all future decisions). The outcome with the highest simulated future rewards can then be deterministically chosen at time t, and this would not decrease the expected total reward of the algorithm. In this way, we can sequentially de-randomize the decision at each time step t. To formalize this argument, we must ensure we draw enough simulation samples to correctly choose the best decision at each time step, and in the end we will lose an $O(\epsilon)$ in the approximation factor. Please see Ma et al. (2021, Sec. 3) which formalizes this argument and achieves pseudo-polynomial running time.

References

Aggarwal G, Goel G, Karande C, Mehta A (2011) Online vertex-weighted bipartite matching and single-bid budgeted allocations. *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, 1253–1264 (SIAM).

Alaei S (2014) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing* 43(2):930–972.

Alaei S, Fu H, Haghpanah N, Hartline J, Malekian A (2012a) Bayesian optimal auctions via multi-to single-agent reduction. *arXiv preprint arXiv:1203.5099* .

Alaei S, Hajiaghayi M, Liaghat V (2012b) Online prophet-inequality matching with applications to ad allocation. *Proceedings of the 13th ACM Conference on Electronic Commerce*, 18–35.

Alijani R, Banerjee S, Gollapudi S, Munagala K, Wang K (2020) Predict and match: Prophet inequalities with uncertain supply. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4(1):1–23.

Alon N, Spencer JH (2016) *The probabilistic method* (John Wiley & Sons).

Amil A, Makhdoumi A, Wei Y (2022) Multi-item order fulfillment revisited: Lp formulation and prophet inequality. *Available at SSRN 4176274* .

Andrews JM, Farias VF, Khojandi AI, Yan CM (2019) Primal–dual algorithms for order fulfillment at urban outfitters, inc. *INFORMS Journal on Applied Analytics* 49(5):355–370.

Aouad A, Sarıtaç Ö (2022) Dynamic stochastic matching under limited time. *Operations Research* 70(4):2349–2383.

Bai Y, El Housni O, Jin B, Rusmevichientong P, Topaloglu H, Williamson D (2022) Fluid approximations for revenue management under high-variance demand: Good and bad formulations. *Available at SSRN* .

Balseiro S, Kroer C, Kumar R (2023) Online resource allocation under horizon uncertainty. *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 63–64.

Besbes O, Sauré D (2014) Dynamic pricing strategies in the presence of demand shifts. *Manufacturing & Service Operations Management* 16(4):513–528.

Border KC (1991) Implementation of reduced form auctions: A geometric approach. *Econometrica: Journal of the Econometric Society* 1175–1187.

Brubach B, Grammel N, Ma W, Srinivasan A (2023) Online matching frameworks under stochastic rewards, product ranking, and unknown patience. *Operations Research* .

Brubach B, Sankararaman KA, Srinivasan A, Xu P (2016) New algorithms, better bounds, and a novel model for online stochastic matching. *24th Annual European Symposium on Algorithms (ESA 2016)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik).

Chawla S, Hartline JD, Malec DL, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *Proceedings of the forty-second ACM symposium on Theory of computing*, 311–320.

Cygan M, Englert M, Gupta A, Mucha M, Sankowski P (2013) Catch them if you can: how to serve impatient users. *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 485–494.

DeValve L, Wei Y, Wu D, Yuan R (2021) Understanding the value of fulfillment flexibility in an online retailing environment. *Manufacturing & Service Operations Management* .

Devanur NR, Hayes TP (2009) The adwords problem: online keyword matching with budgeted bidders under random permutations. *Proceedings of the 10th ACM conference on Electronic commerce*, 71–78.

Dickerson JP, Sankararaman KA, Srinivasan A, Xu P (2021) Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *ACM Transactions on Economics and Computation (TEAC)* 9(3):1–17.

Dubhashi D, Ranjan D (1998) Balls and bins: A study in negative dependence. *Random Structures & Algorithms* 13(5):99–124.

Dunning I, Huchette J, Lubin M (2017) Jump: A modeling language for mathematical optimization. *SIAM review* 59(2):295–320.

Echenique F, Immorlica N, Vazirani VV (2023) Online and matching-based market design. Technical report, Cambridge University Press.

Ehsani S, Hajiaghayi M, Kesselheim T, Singla S (2018) Prophet secretary for combinatorial auctions and matroids. *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms*, 700–714 (SIAM).

Ekbatani F, Feng Y, Niazadeh R (2022) Online resource allocation with buyback: Optimal algorithms via primal-dual. *arXiv preprint arXiv:2210.11570* .

Esfandiari H, Korula N, Mirrokni V (2015) Online allocation with traffic spikes: Mixing adversarial and stochastic models. *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 169–186.

Feldman J, Korula N, Mirrokni V, Muthukrishnan S, Pál M (2009) Online ad assignment with free disposal. *International workshop on internet and network economics*, 374–385 (Springer).

Gamlath B, Kale S, Svensson O (2019) Beating greedy for stochastic bipartite matching. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2841–2854 (SIAM).

Gandhi R, Khuller S, Parthasarathy S, Srinivasan A (2006) Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)* 53(3):324–360.

Guo C, Huang Z, Tang ZG, Zhang X (2021) Generalizing complex hypotheses on product distributions: Auctions, prophet inequalities, and pandora's problem. *Conference on Learning Theory*, 2248–2288 (PMLR).

Huang Z, Lee CS, Lu J, Shu X (2024) Online matching meets sampling without replacement. *arXiv preprint arXiv:2410.06868* .

Huang Z, Shu X (2021) Online stochastic matching, poisson arrivals, and the natural linear program. *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 682–693.

Huang Z, Shu X, Yan S (2022) The power of multiple choices in online stochastic matching. *arXiv preprint arXiv:2203.02883* .

Hwang D, Jaillet P, Manshadi V (2021) Online resource allocation under partially predictable demand. *Operations Research* 69(3):895–915.

Jiang J, Ma W, Zhang J (2022) Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack. *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1221–1246 (SIAM).

Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, 352–358.

Keyvanshokooh E, Shi C, Van Oyen MP (2021) Online advance scheduling with overtime: A primal-dual approach. *Manufacturing & Service Operations Management* 23(1):246–266.

Ma W, Simchi-Levi D (2020) Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios. *Operations Research* 68(6):1787–1803.

Ma W, Simchi-Levi D, Zhao J (2021) Dynamic pricing (and assortment) under a static calendar. *Management Science* 67(4):2292–2313.

Ma W, Xu P, Xu Y (2023) Fairness maximization among offline agents in online-matching markets. *ACM Transactions on Economics and Computation* 10(4):1–27.

Manshadi V, Rodilitz S, Saban D, Suresh A (2022) Online algorithms for matching platforms with multi-channel traffic. *Proceedings of the 23rd ACM Conference on Economics and Computation*, 986–987.

Manshadi VH, Gharan SO, Saberi A (2012) Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research* 37(4):559–573.

McElfresh DC, Kroer C, Pupyrev S, Sodomka E, Sankararaman KA, Chauvin Z, Dexter N, Dickerson JP (2020) Matching algorithms for blood donation. *Proceedings of the 21st ACM Conference on Economics and Computation*, 463–464.

Papadimitriou C, Pollner T, Saberi A, Wajc D (2021) Online stochastic max-weight bipartite matching: Beyond prophet inequalities. *Proceedings of the 22nd ACM Conference on Economics and Computation*, 763–764.

Samuel-Cahn E (1984) Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability* 1213–1216.

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

Shen M, Tang CS, Wu D, Yuan R, Zhou W (2020) Jd. com: Transaction-level data for the 2020 msom data driven research challenge. *Manufacturing & Service Operations Management* .

Simchi-Levi D, Chen X, Bramel J, et al. (2005) The logic of logistics. *Theory, algorithms, and applications for logistics and supply chain management* .

Stein C, Truong VA, Wang X (2020) Advance service reservations with heterogeneous customers. *Management Science* 66(7):2929–2950.

Talluri K, Van Ryzin G (1999) A randomized linear programming method for computing network bid prices. *Transportation science* 33(2):207–216.

Talluri KT, Van Ryzin G (2004) *The theory and practice of revenue management*, volume 1 (Springer).

Truong VA, Wang X (2019) Prophet inequality with correlated arrival probabilities, with application to two sided matchings. *arXiv preprint arXiv:1901.02552* .

Walczak D (2006) Modeling high demand variance in dynamic programming. *Journal of Revenue and Pricing Management* 5:94–101.