Optimal Correlated Rounding for Multi-item Orders in E-commerce Fulfillment

Will Ma
Graduate School of Business, Columbia University, New York, NY 10027, wm2428@gsb.columbia.edu

Problem definition. We study the dynamic fulfillment problem in e-commerce, in which incoming (multi-item) customer orders must be immediately dispatched to (a combination of) fulfillment centers that have the required inventory. The problem takes place over a finite time horizon, representing the duration until the next inventory replenishment.

Methodology/results. A prevailing approach to this problem, pioneered by Jasin and Sinha (2015), is to write a “deterministic” LP that dictates how frequently inventory from each FC (fulfillment center) should be used to fulfill each item, in different orders from different regions. This allows the fulfillment decision for each order to be made independently. However, making this decision for a single multi-item order is still challenging—how can we avoid splitting the order across too many FC’s, while satisfying the inventory frequency constraints? Jasin and Sinha identify this as a correlated rounding problem and propose an intricate rounding scheme, which is suboptimal by a factor of \( \approx \frac{q}{4} \) on a \( q \)-item order.

In this paper we provide to our knowledge the first substantially improved correlated rounding scheme, which is suboptimal by a factor of only \( 1 + \ln(q) \). We provide another scheme for sparse networks, which is suboptimal by a factor of at most \( d \) if each item is stored in at most \( d \) FC’s. We show both of these guarantees to be optimal in terms of the dependence on \( q \) or \( d \), by reducing to the classical Set Cover problem. Our schemes are also simple and fast, based on an intuitive new idea—items wait for FC’s to “open” at random times, but observe them on “dilated” time scales in order to satisfy the inventory frequency constraints.

Managerial implications. We numerically test our new rounding schemes under the same realistics setups as Jasin and Sinha (2015) and find that they improve runtimes, shorten code, and robustly improve performance. Our code is made publicly available.

History: This is version 3, from August 11th 2022. An initial version of this paper appeared in the Manufacturing & Service Operations Management (MSOM) conference Special Interest Group (SIG) for Supply Chain Management, 2022.

1. Introduction

E-commerce has exploded in recent times, achieving unbelievable global scale, unimaginable delivery speed, and unfathomable system complexity. The short-term operations of a typical e-commerce
giant involves pulling inventory from suppliers into its fulfillment centers (FC’s), including retail stores that can also be used to fulfill online orders; waiting for customers to make purchases, which can be influenced by its powerful search/recommendation engine; and finally delivering the goods to the customer’s doorstep, through a flexible transportation system that allows almost any FC in the network to be used for fulfilling demand from any particular region. This paper focuses on the final part of these operations, which is the problem of dynamically dispatching incoming customer orders to FC’s, while treating the inventory replenishment schedule and search ranking/recommendation decisions as exogenous.

This dynamic fulfillment problem is challenging for several reasons. First, decisions must be made with consideration of the future orders to come, since depleting inventories at the wrong places can set off a chain reaction of long-distance and split shipments, as originally demonstrated by Xu et al. (2009). However, due to the uncertainty in future orders, forward-lookingness requires a high-dimensional stochastic dynamic program that is intractable to solve, as noted by Acimovic and Farias (2019). Meanwhile, even a myopic strategy like using the minimum number of FC’s to satisfy each incoming order, without consideration of future orders, can be computationally hard. Finally, the mere scale and speed of the problem restricts us to fast and simple heuristics, with more elaborate optimizations exacerbating the issue of system complexity.

In light of these challenges, a prevailing approach to the dynamic fulfillment problem is LP-based, as pioneered by Jasin and Sinha (2015). In a nutshell, an LP which views the system as deterministic is written, describing inventory levels of every item at every FC, and expected demands at different regions which includes information about items frequently purchased together in the same order. The objective captures fixed shipping costs (mostly dependent on the number of distinct FC’s used to fulfill an order), variable shipping costs (dependent on items and distances), and shortage costs (dependent on penalties paid for orders not fulfilled). The LP is then solved, providing a “master plan” of transporting supply to demand, which prescribes for different orders from different regions, how frequently each FC should be used to fulfill each item in that order. As
orders come up in real-time, Jasin and Sinha (2015) randomly dispatch the items to FC’s, making sure to follow the fulfillment frequencies outlined in the LP’s plan (see Section 5 for details).

Although seemingly uninformed, this randomized fulfillment algorithm is simple and fast, dispatching instantly and not requiring real-time inventory information across the network once the LP solution is given. Under large system scales, it also performs well, in terms of paying variable shipping and shortage costs similar to that of the LP benchmark. However, fixed costs remain a challenge—the problem of minimizing fixed costs for a single order was already difficult, and having to follow the LP’s fulfillment frequencies only introduces additional constraints. The seminal insight from Jasin and Sinha (2015) is that these frequencies are actually helpful—when using them to randomly assign an FC to each item, if positive correlation is induced in the assignments across items, then many items will end up assigned to the same FC, resulting in not many fixed costs being paid. Jasin and Sinha (2015) also derive an intricate method for inducing this correlation, which reduces the fixed costs from the naive independent method by a factor of 4.

Despite its significance and impact on subsequent work (e.g. Lei et al. 2018, 2021, Zhao et al. 2020), to our understanding, the correlation method of Jasin and Sinha (2015) has never been improved in a substantial way, until now. In this paper, we derive a new method (and extension)—based on observing Poisson processes under “dilated” time scales—that is intuitively simple, computationally faster, and achieves the best-possible guarantee (in two different regimes). We focus the rest of the Introduction on the correlated rounding problem identified by Jasin and Sinha (2015) for a single multi-item order, and defer formalities about its implications for the general multi-item dynamic fulfillment problem to Section 5.

1.1. Correlated Rounding Problem of Jasin and Sinha (2015)

Consider a single order (from a particular region at a particular time), consisting of $q$ items. For each item $i = 1, \ldots, q$ in the order, we are given the fraction of time $u_{ki}$ it must be fulfilled from each eligible FC $k = 1, \ldots, K$, with $\sum_k u_{ki} = 1$. For each FC $k$, a fixed cost of $c_k$ is paid if $k$ is used to fulfill any item. The goal is to randomly “round” each item’s probability vector $(u_{ki})_{k=1}^{K}$ to an actual FC for fulfilling item $i$, in a correlated fashion that minimizes the expected fixed costs paid.
Problem 1 (Jasin and Sinha (2015)). Given \( q \) marginal distributions \((u_{k1})_{k=1}^{K}, \ldots, (u_{kn})_{k=1}^{K}\) over a discrete set \( \{1, \ldots, K\} \) and fixed costs \( c_1, \ldots, c_K \), construct jointly-distributed random variables \( Z_1, \ldots, Z_n \) satisfying
\[
\Pr[Z_i = k] = u_{ki} \text{ for all } k \text{ and } i
\]
that minimizes
\[
\sum_{k=1}^{K} c_k \cdot \Pr\left[ \bigcup_{i=1}^{q} (Z_i = k) \right].
\]

(1)

In Problem 1, \( Z_i \in \{1, \ldots, K\} \) denotes the FC used to fulfill item \( i \), and \( \bigcup_{i=1}^{q} (Z_i = k) \) denotes the event that FC \( k \) is used to fulfill any item, in which case its fixed cost \( c_k \) must be paid. Although solving Problem 1 is hard in general, approximate solutions can be derived by observing that
\[
\Pr\left[ \bigcup_{i=1}^{q} (Z_i = k) \right] \geq \max_{i=1}^{q} \Pr[Z_i = k] = \max_i u_{ki}.
\]

(2)

In words, \( \max_i u_{ki} \) is a lower bound on the probability with which FC \( k \) must be used, and hence it suffices to ensure that no FC \( k \) is used too often in comparison to \( \max_i u_{ki} \). Jasin and Sinha (2015) actually focus on deriving the following, which we will call \( \alpha \)-competitive rounding schemes.

Definition 1 (\( \alpha \)-Competitive Rounding Scheme). For \( \alpha \geq 1 \), an \( \alpha \)-competitive (correlated) rounding scheme is a method for constructing random variables \( Z_1, \ldots, Z_n \) satisfying
\[
\Pr[Z_i = k] = u_{ki} \quad \forall i = 1, \ldots, q, k = 1, \ldots, K
\]
\[
\Pr\left[ \bigcup_{i=1}^{q} (Z_i = k) \right] \leq \alpha \cdot \max_{i} u_{ki} \quad \forall k = 1, \ldots, K
\]

(3) (4)
given any \( q \) marginal distributions \((u_{k1})_{k=1}^{K}, \ldots, (u_{kn})_{k=1}^{K}\) over a discrete set \( \{1, \ldots, K\} \).

An \( \alpha \)-competitive rounding scheme provides a solution to Problem 1 that pays at most \( \alpha \) times the optimal cost, due to the lower bound derived in (2). Jasin and Sinha (2015) derive a \( B(q) \)-competitive correlated rounding scheme, where \( B(q) = \frac{(q+1)^2}{4n} \) if \( q \) is odd and \( B(q) = \frac{q+2}{4} \) if \( q \) is even, with function \( B(q) \) growing approximately as \( q/4 \). Meanwhile, it is easy to see that the naive independent rounding scheme is only \( q \)-competitive, which is worse than \( B(q) \) by a factor of approximately 4. Our new result in this paper is a \((1 + \ln(q))\)-competitive rounding scheme, improving the order-dependence on \( q \) entirely, and matching an \( \Omega(\log q) \) lower bound. Moreover, we use similar ideas to derive a \( d \)-competitive rounding scheme when \( d \) is an upper bound on the number of different FC’s that can fulfill an item, which we also show is best-possible. We now describe the main idea behind our new rounding schemes.
1.2. Main Idea behind Rounding Schemes and Analysis

To induce positive correlation in the FC’s assigned across items, we imagine the following process. Each FC is initially closed, and opens at a random time. Items are assigned to the first FC that they see open. Importantly, each item $i$ views the openings of FC’s on its own *dilated* time scale, calibrated so that the probability of it seeing any FC $k$ open first is exactly $u_{ki}$. Because an FC opening early means that it will be seen first by more (but not necessarily all) items, this induces positive correlation in the FC’s assigned across different items.

To make this precise, for each FC $k$, we draw its opening time $E_k$ independently from an Exponential distribution with mean $1/y_k$, where $y_k := \max_i u_{ki}$. We then define the dilated time scale for an item $i$ as: it sees each FC $k$ open at time $\frac{u_{ki}}{y_k} E_k$, which we note is no earlier than $E_k$, since $\frac{u_{ki}}{y_k} \geq 1$. (If $u_{ki} = 0$, then $\frac{u_{ki}}{y_k} E_k = \infty$, and item $i$ never sees FC $k$ open.) The dilated opening times $\frac{u_{ki}}{y_k} E_k$ are Exponentially distributed with means $\frac{u_{ki}}{y_k} \cdot \frac{1}{y_k} = \frac{1}{u_{ki}}$, and independent across $k$. Through the lens of Poisson processes, it is easy to see that the probability of each FC $k$ arriving first into the view of item $i$ is exactly $u_{ki} u_1 + \ldots + u_{K_i} = u_{ki}$, as desired.

The Poisson lens also helps us upper-bound the probability of an FC $k$ getting used at all. Indeed, since an FC $k$ can only be seen at times *later* than $E_k$, it can only get used if it arrives when at least one item is still waiting, an event whose probability is exponentially decaying over time. Unfortunately, random variable $E_k$ is correlated with the latter event, making the analysis complicated. To fix this, we instead consider a related process where FC $k$ is “repeatedly opening” following a Poisson process of rate $y_k$, which allows us to exploit the memoryless property and take an elementary integral to show that the probability of FC $k$ opening is at most $(1 + \ln(q)) y_k$, completing our sketch of why our first rounding scheme is $(1 + \ln(q))$-competitive.

To motivate our second rounding scheme, we note that the preceding analysis is poor when $q$ is enormous, because for a long time at least one item will still be waiting, during which FC openings will result in usage. Therefore, we consider a modified scheme where each FC $k$ is “forced open” at time $1/y_k$, even if $E_k > 1/y_k$. For each item $i$, it will see each FC $k$ forced open at time $\frac{u_{ki}}{y_k} \cdot \frac{1}{y_k} = \frac{1}{u_{ki}}$. 
Therefore, item $i$ will get “force-assigned” by time $\frac{1}{\max_k u_{ki}}$, and all items will be force-assigned by time $\alpha := \frac{1}{\min_i \max_k u_{ki}}$, regardless of how many items there are. Moreover, if $d$ is an upper bound on $|\{k : u_{ki} > 0\}|$, then $\max_k u_{ki} \geq 1/d$ for all $i$, and hence $\alpha \leq d$. The fact that all items are assigned by time $d$ w.p. 1 allows us to show that no FC gets used with probability more than $dy_k$.

However, these forced openings cause each item $i$ to be over-fulfilled from the FC $m(i)$ that it would first see forced open. Therefore, we make a second modification where for each item $i$, if the over-fulfilled FC $m(i)$ were to “naturally” open (i.e. $E_{m(i)} < 1/y_{m(i)}$), then it is hidden from the view of item $i$ (until it is forced open) with some likelihood. This likelihood can be calibrated so that $i$ ends up seeing every FC $k$ open first with probability exactly $u_{ki}$, as desired.

1.3. Summary of Contributions and Outline of Paper

Results for correlated rounding problem. We now list all our results related to Problem 1 and Definition 1, highlighting the virtues of our rounding schemes in relation to Jasin and Sinha (2015), which we refer to as “JS”.

- We derive a $(1 + \ln(q))$-competitive rounding scheme (Subsection 2.1), which is a substantial improvement upon the $\approx q/4$-competitive rounding scheme of JS. We also derive a $d$-competitive rounding scheme (Subsection 2.2), where $d := \max_i |\{k : u_{ki} > 0\}|$ is a sparsity parameter. These guarantees match respective lower bounds of $\Omega(\log q)$ and $d$, as will be shown below.

- Both of our rounding schemes have a runtime of $O(qK)$. By contrast, the rounding scheme of JS has a runtime of $O(q^2K)$, containing a loop that is quadratic in the number of items $q$.

- Our rounding schemes are intuitive—FC’s have random opening times, and items are assigned to the first FC they see open on a dilated time scale. By contrast, the method of JS based on constructing line partitions, while clever and beautiful, is to our understanding not simple.

- We should acknowledge that our guarantee is $1 + \ln 2 \approx 1.69$ when $q = 2$. By contrast, JS is 1-competitive if $q = 2$. We also note that if there are only two FC’s, i.e. $K = 2$, then a 1-competitive rounding scheme was recently discovered by Zhao et al. (2020). In this scenario, our second rounding scheme would only be 2-competitive, since $d = K = 2$. However, we emphasize that
parameter $d$ represents the maximum number of distinct FC’s that hold an item and can generally be much smaller than $K$, whereas their rounding scheme only works when $K = 2$.

- As an additional result, we show how Problem 1 can be solved optimally using an LP of size $O(2^K)$ (Section 4). JS also have a result that solves Problem 1 optimally using an LP of size $O(K^q)$. While both of these LP’s are exponentially-sized, our LP can be applied when $K$ is a small, while their LP can be applied when $q$ is a small.

We also consider the overall multi-item dynamic fulfillment problem of Jasin and Sinha (2015) with multiple query types, regions, and time steps (Section 5), numerically testing our rounding schemes on realistic instances of it constructed in the same way as JS. As shown in Section 6, building upon the randomized fulfillment framework of Jasin and Sinha (2015) but using our new idea of dilated opening times to do correlated rounding instead, we achieve faster computation, much simpler code, and robustly better performance across a variety of experimental setups. Our code is also made publicly available so that our findings can be replicated.

**Relations with set cover rounding problem.** We make further contributions by relating $\alpha$-competitive rounding schemes to notions from the Set Cover problem, establishing the following.

- We show that an $\alpha$-competitive rounding scheme implies a procedure for rounding a fractional Set Cover solution into a randomized cover, that is feasible w.p. 1, and having no set chosen with probability more than $\alpha$ times its fractional weight (Section 3).

- Therefore, we can leverage hardness results from Set Cover to show that an $\alpha$-competitive rounding scheme must have $\alpha = \Omega(\log q)$ and $\alpha \geq d$ (Subsection 3.1). The latter lower bound establishes our $d$-competitive rounding scheme to be exactly tight, not just order-optimal.

- Our $(1 + \ln(q))$-competitive rounding scheme also improves existing guarantees in the aforementioned randomized rounding problem for Set Cover. Existing methods need to select sets with probability at least $2\ln(q)$ times their fractional weight; see Yazirani (2001, Sec 14.2), Buchbinder et al. (2009, Sec. 2.2.2), or Motwani and Raghavan (1995). The key is that our method induces sets to be selected in a *negatively correlated* fashion, whereas existing methods select sets *independently* and only show that the solution is feasible with high probability.
We note that for the Set Cover problem itself (not its rounding problem), our results do not improve existing guarantees—the Greedy algorithm already has a guarantee of \(1 + 1/2 + \cdots + 1/q\) which is smaller (better) than our \(1 + \ln(q)\). Nonetheless, we believe these connections highlight how the correlated rounding problem is a harder version of Set Cover—in which a randomized solution, that must satisfy constraints on how often each set is used to cover each element, is required. Furthermore, related to the final bullet above, it is interesting to us that a modern problem from e-commerce practice, identified by Jasin and Sinha (2015), can lead us to improve randomized rounding schemes for the age-old Set Cover problem from CS theory.

1.4. Further Related Work

We briefly mention some further related work that helps justify our approach.

First, we note that the problem we solve is highly relevant, especially for omni-channel retailers who could potentially fulfill from hundreds of retail stores (Acimovic and Farias 2019). The number of items in an order can also be large, with online retailers taking steps to consolidate multiple orders into one big order to be fulfilled together (Wei et al. 2021). Finally, the fact that the e-tailer wants flexibility when dynamically deciding fulfillment breakdowns is justified in DeValve et al. (2021).

In terms of the overall LP-based approach that justifies the correlated rounding problem, we should note that LP-based approaches are also heavily employed in the revenue management literature (see e.g. Talluri and Van Ryzin 2004). They enjoy many benefits such as scalability and ability to incorporate side constraints, and the given probabilities \(u_{ki}\) can always be updated over time through re-solving (see e.g. Jasin and Kumar 2012) to adjust for updated inventories and demand predictions over time. Another early work that discusses the benefits of using LP-based approaches to look ahead in e-commerce fulfillment is Acimovic and Graves (2015).

2. Definition and Analysis of Rounding Schemes

We now provide efficient algorithmic specifications of our rounding schemes and analyze them. We believe both our algorithms and proofs to be quite intuitive, and will frequently provide proof
sketches that refer back to the intuition from Subsection 1.2, where items are waiting for FC’s to open on their own dilated time scales.

Before proceeding, we recap the problem and the notation/terminology to be used.

**Definition 2 (Recap of Problem, Notation, and Terminology).**

- An *instance* of the $\alpha$-competitive rounding scheme problem consists of $q$ marginal distributions over $K$ FC’s, given by probabilities $u_{ki}$ satisfying $\sum_{k=1}^{K} u_{ki} = 1$ for all $i = 1, \ldots, q$.
- A *rounding scheme* must randomly assign each item $i$ to an FC $Z_{i} \in \{1, \ldots, K\}$, satisfying the marginal conditions $\Pr[Z_{i} = k] = u_{ki}$ for all $i$ and $k$.
- An FC $k$ is *used* if any item is assigned to it, which must occur with probability at least $y_{k} := \max_{i} u_{ki}$. Assume without loss of generality that $y_{k} > 0$ for all $k$.
- A rounding scheme is *$\alpha$-competitive* if given any instance, it uses each FC $k$ with probability at most $\alpha y_{k}$. The guarantee $\alpha$ can depend on parameters of the instance.
- The *sparsity* of an instance is defined as $d = \max_{i} |\{k : u_{ki} > 0\}|$, the maximum number of distinct FC’s that one item $i$ could get assigned to.

### 2.1. $(1 + \ln(q))$-competitive Rounding Scheme

Our rounding scheme is specified in Algorithm 1. Relating back to the intuitive description, $E_{k}$ is the time at which FC $k$ opens, and $\frac{w_{k}}{u_{ki}} E_{k}$ is the delayed time (since $\frac{w_{k}}{u_{ki}} \geq 1$) at which item $i$ sees it open, with $\frac{w_{k}}{u_{ki}} E_{k} = \infty$ if $u_{ki} = 0$. Every item is assigned to the first FC that it sees open.

**Algorithm 1 $(1 + \ln(q))$-competitive Rounding Scheme**

```plaintext
for $k = 1, \ldots, K$ do
    $E_{k} \leftarrow \text{independent draw from Exponential distribution with mean } 1/y_{k}$
end for

for $i = 1, \ldots, q$ do
    $Z_{i} \leftarrow \text{arg min}_{k=1,\ldots,K} \frac{w_{k}}{u_{ki}} E_{k}$  \text{▷}  \text{Break ties arbitrarily}
end for
```
We now prove that Algorithm 1 is a \( (1 + \ln(q)) \)-competitive Rounding Scheme, where \( q \) is the number of items. To establish the marginals condition, we use the interpretation that from the perspective of any individual item, the FC’s open according to independent Poisson processes.

**Lemma 1.** Under Algorithm 1, \( \Pr[Z_i = k] = u_{ki} \) for all \( i = 1, \ldots, q \) and \( k = 1, \ldots, K \).

**Proof of Lemma 1.** Consider the perspective of any item \( i \). Index \( Z_i \) is determined by the smallest realization among \( \{ \frac{1}{u_{ki}} E_k : k = 1, \ldots, K \} \), which are independent Exponential random variables with means \( \{ \frac{1}{u_{ki}} : k = 1, \ldots, K \} \). Equivalently, \( Z_i \) is determined by the first arrival among independent Poisson processes with rates \( \{ u_{ki} : k = 1, \ldots, K \} \). By the Poisson merging theorem, each Poisson process \( k \) will be the first to arrive with probability \( \frac{u_{ki}}{u_{1i} + \ldots + u_{Ki}} \), which equals \( u_{ki} \) since \( u_{1i} + \ldots + u_{Ki} = 1 \). Therefore, \( \Pr[Z_i = k] = u_{ki} \) for all \( k = 1, \ldots, K \), completing the proof. \( \square \)

We now prove an intermediate lemma that, intuitively, bounds the probability of any item \( i \) still “waiting” (to be assigned to an FC) up to time \( t \), which can be expressed as the event \( (\min_k \frac{u_{ki}}{u_{ki}} E_k \geq t) \). The final statement then takes a union bound of having any item still waiting, which intuitively is not too loose since these events are positively correlated—one item waiting implies that FC’s were late to open, which makes other items more likely to also be waiting.

**Lemma 2.** Under Algorithm 1, \( \Pr[\bigcup_{i=1}^q (\min_k \frac{u_{ki}}{u_{ki}} E_k \geq t)] \leq ne^{-t} \) for all \( t \geq 0 \).

**Proof of Lemma 2.** First consider any item \( i \). Random variables \( \{ \frac{u_{ki}}{u_{ki}} E_k : k = 1, \ldots, K \} \) are independent and Exponentially distributed with means \( \{ \frac{1}{u_{ki}} : k = 1, \ldots, K \} \). Therefore, \( \min_k \frac{u_{ki}}{u_{ki}} E_k \) is Exponentially distributed with mean \( \frac{1}{u_{1i} + \ldots + u_{Ki}} = 1 \). Consequently, \( \Pr[\min_k \frac{u_{ki}}{u_{ki}} E_k \geq t] = e^{-t} \), and by the union bound, \( \Pr[\bigcup_{i=1}^q (\min_k \frac{u_{ki}}{u_{ki}} E_k \geq t)] \leq ne^{-t} \), completing the proof. \( \square \)

We are now ready to prove our main result for Algorithm 1. Although technical, the argument uses a simple intuitive trick. Lemma 2 has upper-bounded the probability of any item still waiting at a time \( t \). If an FC \( k \) opens at a time when no item is still waiting, then it is guaranteed to not get used (since items can only see it open at a **delayed** time). Unfortunately, the opening time of an FC \( k \) is correlated with the event of having an item still waiting. To fix this, we imagine FC \( k \)
as “repeatedly opening” following a Poisson process of rate $y_k$, with it being “used” every time it opens as long as there is an item still waiting. Since Poisson processes are memoryless, this now de-correlates the events of FC $k$ opening from the event of still having an item waiting. Lemma 2 can then apply, and the analysis finishes by taking an integral. The formal proof is presented below.

**Theorem 1.** Algorithm 1 is a $(1 + \ln(q))$-competitive rounding scheme with runtime $O(qK)$.

**Proof of Theorem 1.** The runtime is $O(qK)$ because taking the arg min over $k = 1, \ldots, K$ for all $i = 1, \ldots, q$ is the bottleneck operation in Algorithm 1. Meanwhile, Lemma 1 has already shown that the marginals condition is satisfied. It remains to show that $Pr[\bigcup_{i=1,\ldots,q} (Z_i = k)] \leq \alpha y_k$ for all $k$, with $\alpha = 1 + \ln(q)$.

Fix any FC $k$. For all items $i$ with $u_{ki} > 0$, event $Z_i = k$ can occur only if $k$ lies in the arg min in Algorithm 1, i.e., if $\min_{k'} \frac{y_{ki}}{u_{ki}} E_{k'} \geq \frac{y_k}{u_{ki}} E_k$. We now rewrite this event as follows. Define $S_k^1, S_k^2, \ldots$ to be the arrival times of a Poisson process of rate $y_k$. More specifically, we will let $S_k^j = E_k$, and $S_k^{j+1}$ be the sum of $S_k^j$ with an independent Exponential random variable of mean $1/y_k$, for all $j \geq 1$. We can technically derive

\[(Z_i = k) \subseteq \left( \min_{k'} \frac{y_{ki}}{u_{ki}} E_{k'} \geq \frac{y_k}{u_{ki}} E_k \right) \]

\[= \left( \min_{k' \neq k} \frac{y_{ki}}{u_{ki}} E_{k'} \geq \frac{y_k}{u_{ki}} S_k^1 \right) \]

\[= \bigcup_{j=1}^{\infty} \left( \min \left\{ \min_{k' \neq k} \frac{y_{ki}}{u_{ki}} E_{k'}, \min_{j' < j} \frac{y_k}{u_{ki}} S_k^{j'} \right\} \geq \frac{y_k}{u_{ki}} S_k^j \right) \] (5)

We now take a union bound of events (5) over $i$, and analyze the probability of this union by conditioning on the event that $S_k^j = t$ for any $j \geq 1$, over all times $t \geq 0$. Formally:

\[
\Pr \left[ \bigcup_{i : u_{ki} > 0} \bigcup_{j=1}^{\infty} \left( \min \left\{ \min_{k' \neq k} \frac{y_{ki}}{u_{ki}} E_{k'}, \min_{j' < j} \frac{y_k}{u_{ki}} S_k^{j'} \right\} \geq \frac{y_k}{u_{ki}} S_k^j \right) \right]
\]

\[= \int_0^{\infty} \Pr \left[ \bigcup_{i : u_{ki} > 0} \left( \min \left\{ \min_{k' \neq k} \frac{y_{ki}}{u_{ki}} E_{k'}, \min_{j' < j} \frac{y_k}{u_{ki}} S_k^{j'} \right\} \geq \frac{y_k}{u_{ki}} t \right) \right] y_k dt
\]

\[\leq \int_0^{\infty} \Pr \left[ \bigcup_{i : u_{ki} > 0} \left( \min \left\{ \min_{k' \neq k} \frac{y_{ki}}{u_{ki}} E_{k'}, \min_{j' < j} \frac{y_k}{u_{ki}} S_k^{j'} \right\} \geq t \right) \right] y_k dt
\]
\[
= \int_0^\infty \Pr \left[ \bigcup_{\cup u_{ki} > 0} \min_{k' = 1, \ldots, K} \frac{y_{k'}}{u_{ki}} E_{k'} \geq t \right] y_k dt \\
\leq y_k \int_0^\infty \min\{ ne^{-t}, 1\} dt
\]
where the first equality holds because the PDF of the event \( \exists j : S_j^i = t \) takes value \( y_k \) for all \( t \), the first inequality holds because \( \frac{y_k}{u_{ki}} \geq 1 \), the second equality applies the memorylessness property of Poisson processes, and the final inequality applies Lemma 2 (along with the trivial upper bound of 1). Note that this analysis holds for any FC \( k = 1, \ldots, K \). Therefore, the proof is now completed by taking an elementary integral:

\[
\int_0^\infty \min\{ ne^{-t}, 1\} dt = \ln(q) + \int_0^\infty ne^{-t} dt \\
= \ln(q) + ne^{-\ln(q)} \\
= 1 + \ln(q). \quad \square
\]

2.2. \( d \)-competitive Rounding Scheme

Our modified rounding scheme is specified in Algorithm 2. Relating back to the intuitive description from Subsection 1.2, \( m(i) \) is the first FC that item \( i \) would see “forced” open, which it would get assigned to if it was still unassigned at that point. \( X_{ki} \) is a random variable denoting the time at which item \( i \) sees FC \( k \) open, which equals \( u_{ki} E_k \) like before if \( k \neq m(i) \). On the other hand, \( X_{m(i),i} \) is upper-bounded by \( 1/u_{m(i),i} \), as that is when item \( i \) would see FC \( m(i) \) forced open. The final wrinkle is that if FC \( m(i) \) were to “naturally” open before it is forced open, then it needs to be hidden from \( i \)'s view (until it is forced open) with some probability, which is indicated by the random variable \( H_i \). Finally, every item is assigned to the first FC that it sees open, after taking into consideration hiding and forced opening.

It can be checked that the probability with which \( H_i = 1 \) defined in Algorithm 2 does indeed lie in \([0,1]\) for all possible values of \( u_{m(i),i} \in (0,1] \). The hiding probability is in fact increasing in \( u_{m(i),i} \), which is intuitive because a larger value of \( u_{m(i),i} \) implies an earlier forced opening, suggesting that FC \( m(i) \) should be hidden more often to prevent it from over-fulfilling item \( i \). We now prove that this hiding probability has been calibrated so that the marginals condition is satisfied exactly.
**Algorithm 2** $d$-competitive Rounding Scheme

for $k = 1, \ldots, K$ do

\[ E_k \leftarrow \text{independent draw from Exponential distribution with mean } \frac{1}{y_k} \]

end for

for $i = 1, \ldots, q$ do

\[ m(i) \leftarrow \arg \max_k u_{ki} \]

for $k = 1, \ldots, K$, $k \neq m(i)$ do

\[ X_{ki} \leftarrow \frac{y_k}{u_{ki}} E_k \]

end for

$H_i \leftarrow \text{independent draw from Bernoulli distribution with mean } \frac{1 - u_{m(i),i}}{1 - u_{m(i),i} + u_{m(i),i} e^{1/y_{m(i),i} - e}}$

\[ \triangleright H_i = 1 \text{ means FC } m(i) \text{ is hidden from item } i \text{ until the FC is forced open at time } \frac{1}{y_{m(i)}} \]

\[ X_{m(i),i} \leftarrow \frac{y_{m(i)}}{u_{m(i),i}} \min \left\{ \frac{E_{m(i)}}{1 - H_i}, \frac{1}{y_{m(i)}} \right\} \quad \triangleright H_i = 1 \text{ means } \frac{E_{m(i)}}{1 - H_i} = \infty, \text{ and hence } X_{m(i),i} = \frac{1}{u_{m(i),i}} \]

$Z_i \leftarrow \arg \min_{k=1,\ldots,K} X_{ki}$

end for

**Lemma 3.** Under Algorithm 2, $\Pr[Z_i = k] = u_{ki}$ for all $i = 1, \ldots, q$ and $k = 1, \ldots, K$.

**Proof of Lemma 3.** Fix any item $i$. We show that $\Pr[Z_i = k] = u_{ki}$ for all $k \neq m(i)$, which would automatically imply $\Pr[Z_i = m(i)] = 1 - \sum_{k \neq m(i)} \Pr[Z_i = k] = 1 - \sum_{k \neq m(i)} u_{ki} = u_{m(i),i}$. We need to consider two cases: $H_i = 1$ and $H_i = 0$. Hereafter omit index $i$.

First, if $H = 1$, then the item does not observe FC $m$ before time $1/u_m$. Therefore, $Z = k$ if and only if $X_k$ is the smallest among random variables $X_{k'} : k' \neq m$ and also $X_k < 1/u_m$. Recall that $X_{k'}$ is Exponentially distributed with mean $1/u_{k'}$ for all $k' \neq m$, and the $X_{k'}$’s are independent across $k'$. Therefore, the probability that $\min_{k' \neq m} X_{k'} < 1/u_m$ is equal to the probability that a Poisson process with rate $\sum_{k' \neq m} u_{k'} = 1 - u_m$ generates an arrival before time $1/u_m$, which occurs w.p. $1 - e^{-(1 - u_m)/u_m}$. Conditional on this, the probability that $\min_{k' \neq m} X_{k'} = X_k$ is exactly $\frac{u_k}{1 - u_m}$, by the Poisson merging theorem. Therefore,

\[ \Pr[Z = k | H = 1] = (1 - e^{-(1 - u_m)/u_m}) \frac{u_k}{1 - u_m}. \quad (6) \]
Otherwise, if $H = 0$, then the item observes all FC’s before time $1/u_m$. In this case, $Z = k$ if and only if $X_k$ is the smallest among all random variables $\{X_{k'} : k' = 1, \ldots, K\}$ and also $X_k < 1/u_m$. By a similar argument as above, the probability that $\min_{k'=1,\ldots,K} X_{k'} < 1/u_m$ is $1 - e^{1/u_m}$, and conditional on this, the probability that $\min_{k'=1,\ldots,K} X_{k'} = X_k$ is $u_k$. Therefore,

$$\Pr[Z = k | H = 0] = (1 - e^{1/u_m}) u_k. \quad (7)$$

Let $\eta$ denote $\frac{1-u_m}{1-u_m + u_m e^{1/u_m} - e}$, the probability that $H = 1$. Combining (6) and (7), we derive

$$\Pr[Z_i = k] = \eta (1 - e^{-(1-u_m)/u_m}) \frac{u_k}{1-u_m} + (1-\eta) (1 - e^{-1/u_m}) u_k$$

$$= u_k \left( 1 - e^{-1/u_m} + \eta \left( \frac{1 - e^{-(1-u_m)/u_m}}{1-u_m} - (1 - e^{-1/u_m}) \right) \right)$$

$$= u_k \left( 1 - e^{-1/u_m} + \eta \cdot \frac{e^{-(1-u_m)/u_m} + u_m + e^{-1/u_m} - u_m e^{-1/u_m}}{1-u_m} \right)$$

$$= u_k \left( 1 - e^{-1/u_m} + e^{-1/u_m} \eta \cdot \frac{1 - u_m + u_m e^{1/u_m} - e}{1-u_m} \right)$$

which completes the proof.  \(\square\)

We now prove our main result for Algorithm 2. We establish the stronger guarantee of $\alpha = \frac{1}{\min_i \max_k u_{ki}}$, which is easily seen to be at most $d$ since $\max_k u_{ki} \geq 1/d$ for all $i$. The proof sketch is that due to the forced openings, all items are guaranteed to be assigned by time $\alpha$. Therefore, an FC $k$ can only get used is it opens before time $\alpha$ (since items can only see it open with a delay), which occurs with probability no greater than $\alpha y_k$.

**Theorem 2.** Algorithm 2 is an $\frac{1}{\min_i \max_k u_{ki}}$-competitive rounding scheme with runtime $O(qK)$.

**Proof of Theorem 2.** The runtime is $O(qK)$, because inside the loop for $i = 1, \ldots, q$ in Algorithm 2 there are three bottleneck operations that each take time $O(K)$: the defining of $m(i)$, the inner loop for $k$, and the defining of $Z_i$. Meanwhile, Lemma 3 has already shown that the marginals condition is satisfied. It remains to show that $\Pr[\bigcup_{i=1,\ldots,q} (Z_i = k)] \leq \alpha y_k$ for all $k$, with $\alpha = \frac{1}{\min_i \max_k u_{ki}}$. 


Fix an FC $k$. If $y_k \geq \min_i \max_{k'} u_{k'i}$, then $\alpha y_k \geq 1$ and there is nothing to prove. Therefore, assume $y_k < \min_i \max_{k'} u_{k'i}$, and we must show that $\Pr[\bigcup_{i=1}^q (Z_i = k)] \leq \alpha y_k$. Since $y_k < \max_{k'} u_{k'i}$ for all $i$, we know that $k \neq m(i)$ for all $i$. Thus, we have $X_{ki} = \frac{y_k}{u_{ki}} E_k$ for all $i$, and can write

$$(Z_i = k) \subseteq \left( \frac{y_k}{u_{ki}} E_k \leq \min_{k'=1,\ldots,K} X_{k'i} \right)$$

$$\subseteq \left( \frac{y_k}{u_{ki}} E_k \leq X_{m(i),i} \right)$$

$$\subseteq \left( \frac{y_k}{u_{ki}} E_k \leq 1/u_{m(i),i} \right)$$

$$= \left( \frac{y_k}{u_{ki}} E_k \leq \frac{1}{\max_{k'} u_{k'i}} \right)$$

$$\subseteq \left( \frac{y_k}{u_{ki}} E_k \leq \alpha \right)$$

$$\subseteq \left( E_k \leq \alpha \right)$$

with the final relationship between events holding because $\frac{y_k}{u_{ki}} \geq 1$. Note that the final event is independent of $i$. Therefore,

$$\Pr \left[ \bigcup_{i=1}^q (Z_i = k) \right] \leq \Pr [E_k \leq \alpha] = 1 - e^{-\alpha y_k}$$

which is at most $\alpha y_k$, completing the proof. □

3. Connections with Set Cover

In this section we establish our rounding schemes to be order-optimal in terms of the dependence on $q$ or $d$, by reducing our problem to that of rounding a fractional solution for Set Cover. We first define the Set Cover problem and some basic concepts using our language of items and FC’s. We refer to Vazirani (2001) for further background.

Problem 2 (Weighted Set Cover). There are items $i = 1, \ldots, q$ to be covered by FC’s $k = 1, \ldots, K$. Each FC $k$ requires a fixed cost of $c_k$ to open, and if opened, can cover all items in a set $U_k \subseteq \{1, \ldots, q\}$. The objective is to find a collection of FC’s to open, that covers all the items, and minimizes the sum of fixed costs paid for opening FC’s. The sparsity of the instance is defined as $d := \max_i |\{k : i \in U_k\}|$, the maximum number of different FC’s that an item $i$ can be covered by.
Definition 3 (Set Cover Linear/Integer Programs). The following Integer Program is called the Set Cover IP. In it, binary variable $y_k$ represents FC $k$ being opened. It is an equivalent formulation of the Weighted Set Cover problem.

\[
\begin{align*}
\min & \quad \sum_{k=1}^{K} c_k y_k \\
\text{s.t.} & \quad \sum_{k : i \in U_k} y_k \geq 1 & \forall i = 1, \ldots, q \\
& \quad y_k \in \{0, 1\} & \forall k = 1, \ldots, K
\end{align*}
\] (8)

(9)

Meanwhile, the Set Cover LP is defined as the relaxation of the Set Cover IP with constraint (9) changed to $y_k \in [0, 1]$, for all $K = 1, \ldots, K$.

We now define the problem of rounding a fractional solution for Set Cover, in a way analogous to how we defined $\alpha$-competitive rounding scheme in Definition 1, except we will call it an $\alpha$-competitive “covering” scheme instead.

Definition 4 (\(\alpha\)-Competitive Covering Scheme). For $\alpha \geq 1$, an $\alpha$-competitive covering scheme is a method for constructing random variables $Y_1, \ldots, Y_K \in \{0, 1\}$ satisfying

\[
\begin{align*}
\sum_{k : i \in U_k} Y_k \geq 1 & \quad \forall i = 1, \ldots, q, \text{ w.p. } 1 \\
\mathbb{E}[Y_k] & \leq \alpha \cdot y_k & \forall k = 1, \ldots, K
\end{align*}
\] (10)

(11)

given any feasible solution $(y_k)_{k=1}^{K}$ to the Set Cover LP.

We now show that coming up with $\alpha$-competitive rounding schemes is a harder problem than coming up with $\alpha$-competitive covering schemes.

Lemma 4. An $\alpha$-competitive rounding scheme can be efficiently applied as an $\alpha$-competitive covering scheme. Moreover, any dependence of $\alpha$ on the parameters $q$ or $d$ translate over directly.

Proof of Lemma 4. Take any instance of Set Cover and a feasible solution $(y_k)_{k=1}^{K}$ to its LP. For each item $i$, arbitrarily set $u_{ki} \in [0, y_k]$ for each FC $k$ that can cover it, so that $\sum_{k : i \in U_k} u_{ki} = 1$. 
We note that this is always possible since \( y_k \geq 0 \) and \( \sum_{k:i \in U_k} y_k \geq 1 \) by (8). Meanwhile, set \( u_{ki} = 0 \) if \( i \notin U_k \).

The marginal distributions \((u_k)_{k=1}^K, \ldots, (u_{kn})_{k=1}^K\) now define an instance for Definition 1 with the same number of items \( q \) and a sparsity \( d \) that is no greater than before. We apply the \( \alpha \)-competitive rounding scheme that is assumed to exist on this instance, and define random variables 
\[
Y_k = 1(\bigcup_i (Z_i = k)) \text{ for all } k = 1, \ldots, K.
\]
By condition (3) for the rounding scheme, for each item \( i \), we know that \( Z_i = k \) is true for some index \( k \in \{1, \ldots, K\} \), with \( k \in U_k \) since otherwise \( u_{ki} = 0 \). Therefore, \( Y_k = 1 \) for this index \( k \) and condition (10) for the covering scheme is satisfied. Meanwhile, applying condition (4) for the rounding scheme, we have
\[
E[Y_k] = \Pr[\bigcup_i (Z_i = k)] \leq \alpha \cdot \max_i u_{ki} \leq y_k.
\]
We conclude that condition (11) for the covering is satisfied. We also note that if \( \alpha \) depends on the sparsity parameter \( d \), then the same guarantee continues to hold under the old sparsity parameter for Set Cover which is no less than \( d \), completing the proof. \( \square \)

3.1. Negative Results for \( \alpha \)-competitive Rounding Schemes

Equipped with Lemma 4, we can now translate hardness results for the \( \alpha \)-competitive covering scheme problem into hardness results for the \( \alpha \)-competitive rounding scheme problem.

**Corollary 1 (of Lemma 4).** An \( \alpha \)-competitive covering scheme must have \( \alpha = \Omega(\log q) \) (Vazirani 2001, Ex. 13.4). Therefore, an \( \alpha \)-competitive rounding scheme must also have \( \alpha = \Omega(\log q) \). Consequently, the \((1 + \ln(q))\)-competitive rounding scheme established in Theorem 7 achieves the order-optimal dependence on \( q \).

**Proposition 1.** An \( \alpha \)-competitive covering scheme must have \( \alpha \geq d \), where \( d \) denotes the sparsity of the instance.

**Proof of Proposition 1.** Consider a Set Cover instance with \( d \) fixed, \( K \) large, and one item for each subset of \( \{1, \ldots, K\} \) of size \( d \). Each such item can only be covered by the \( d \) FC’s in its
corresponding subset, with the total number of items being \( q = \binom{K}{d} \). The sparsity of this instance is \( d \) by definition.

Setting \( y_k = 1/d \) for all \( k = 1, \ldots, K \) forms a feasible solution to the Set Cover LP, since \( |\{k : i \in U_k\}| = d \) for all items \( i \), and hence LP constraints (8) are satisfied. On the other hand, any \( \alpha \)-competitive covering scheme must set \( \sum_{k=1}^{K} Y_k > K - d \) w.p. 1, since otherwise there would be an uncovered item, violating (10). Using the linearity of expectation, we derive

\[
K - d \leq \sum_{k=1}^{K} \mathbb{E}[Y_k] \leq \sum_{k=1}^{K} \alpha \cdot y_k = K \alpha \frac{1}{d},
\]

with the second inequality coming from (11). Therefore, \( \alpha \geq d(1 - \frac{d}{K}) \), with \( \frac{d}{K} \) approaching for arbitrarily large \( K \), completing the proof. \( \square \)

**Corollary 2 (of Lemma 4 and Proposition 1).** An \( \alpha \)-competitive rounding scheme must have \( \alpha \geq d \). Consequently, the \( d \)-competitive rounding scheme established in Theorem 2 achieves the optimal (not just order-optimal) dependence on \( d \).

### 4. Instance-Optimal Rounding Schemes

The \((1 + \ln(q))\)- and \( d \)-competitive rounding schemes presented in Section 2 are only (order-) optimal in the worst case. For a particular instance given by \( q \) marginals over \( \{1, \ldots, K\} \), one could try to compute the maximum feasible value of \( \alpha \) in Definition 1 for that instance, or solve Problem 1 exactly. We now provide a method for doing either of these two tasks.

Our method is based on a new LP with the following variables. For all subsets \( S \) of the FC’s \( \{1, \ldots, K\} \), let \( z(S) \) denote the probability that exactly the set of FC’s in \( S \) get used. For all \( S \subseteq \{1, \ldots, K\} \), FC’s \( k \in S \), and items \( i \), let \( u_{ki}(S) \) denote the probability that the set of FC’s in \( S \) get used and item \( i \) is fulfilled from FC \( k \in S \). Problem 1 for minimizing fixed costs can then be formulated as

\[
\min \sum_{S} z(S) \sum_{k \in S} c_k \tag{12}
\]

subject to

\[
\sum_{k \in S} u_{ki}(S) = z(S) \quad \forall S, i = 1, \ldots, q \tag{13}
\]
\[ \sum_S u_{ki}(S) = u_{ki} \quad k = 1, \ldots, K, i = 1, \ldots, q \quad (14) \]

\[ \sum_S z(S) = 1 \quad (15) \]

\[ z(S) \geq 0 \quad \forall S \quad (16) \]

\[ u_{ki}(S) \geq 0 \quad \forall S, k \in S, i = 1, \ldots, q \quad (17) \]

where constraints (13) enforce that every item \( i \) must be fulfilled from exactly one FC on each subset \( S \), constraints (14) enforce the marginals condition, constraints (15)–(16) enforce that exactly one subset \( S \) is selected, and last but not least, (17) ensures that there is only a variable \( u_{ki}(S) \) if \( k \in S \).

The LP does not enforce that every FC \( k \in S \) actually gets used (i.e. has \( u_{ki}(S) > 0 \) for some \( i \)), but note that if not, then \( k \) can be discarded from the set \( S \) while decreasing fixed costs.

To see that our LP can also model the problem of finding the maximum feasible value of \( \alpha \) in Definition 1, we simply have to replace the objective function (12) with \( \min \alpha \) and add constraints

\[ \sum_{S \ni k} z(S) \leq y_k \alpha \quad \forall k = 1, \ldots, K \quad (18) \]

where \( y_k = \max_i u_{ki} \) and \( \alpha \) is an additional variable.

Both of our LP’s have size \( O(nK^2K^q) \), which is exponential in \( K \) but tractable if \( K \) is a fixed constant. Jasin and Sinha (2015) derive LP’s for the same purposes, except instead there is a variable for every possible mapping from \( \{1, \ldots, q\} \) to \( \{1, \ldots, K\} \), for which there are \( K^q \) possibilities. Our LP’s are more practical in situations where \( K \) is small but \( q \) is large, which is the case in the application of e.g. Zhao et al. (2020).

5. \( \alpha \)-competitive Rounding Scheme applied to Dynamic Fulfillment

In this section we recap the general dynamic fulfillment problem from Jasin and Sinha (2015), and formalize the implication of our \( \alpha \)-competitive rounding schemes for the overall problem.

**Problem definition.** There is a horizon consisting of time steps \( t = 1, \ldots, T \), during which items \( i = 1, \ldots, n \) are fulfilled from FC’s \( k = 1, \ldots, K \). Each item \( i \) starts with \( b_{ki} \) units of inventory at each
FC \( k \), with the end of the horizon representing the time at which inventories are replenished again. Orders come from one of regions \( j = 1, \ldots, J \), and are described by a subset \( q \) of items \( q \subseteq \{1, \ldots, n\} \) that was just purchased. During each time step, up to one order arrives, which is from region \( j \) and is for subset \( q \) with probability \( \lambda^q_j \), with \( \sum_{q,j} \lambda^q_j \leq 1 \). As is standard in revenue management, we assume a granular division of time such that at most one order can arrive during each time step. Also, as justified in Jasin and Sinha (2015), we assume that orders cannot contain more than one of any item, and assume a small universe of possible subsets \( q \). We let \( c^\text{unit}_{kij} \) denote the variable cost of fulfilling one unit of item \( i \) from FC \( k \) to location \( j \), and let \( c^\text{fixed}_{kj} \) denote the fixed cost of sending a package (containing one or more items) from FC \( k \) to location \( j \).

The goal is to dynamically decide the FC’s to use to fulfill the items in each order that arrives over the time horizon, to minimize total expected cost. Note that if an FC \( k \) is used to fulfill a subset \( q' \subseteq q \) of an order from a location \( j \), then the cost required to send that package is \( c^\text{fixed}_{kj} + \sum_{i \in q'} c^\text{unit}_{kij} \). All items in each arriving order must be fulfilled from some FC, where we assume the existence of a null FC 0 with infinite inventory so that this is always feasible, with \( c^\text{unit}_{0ij} \) denoting the “shortage” cost of failing to fulfill one unit of item \( i \) to region \( j \).

**LP benchmark.** Solving for the optimal dynamic fulfillment policy using dynamic programming is intractable, since the state space is exponential in the number of items. Thus, the following “deterministic” LP benchmark is often used to derive heuristic policies and bound their suboptimality relative to the optimal dynamic programming policy.

\[
\text{DLP} := \min \sum_{q,k,j} T \lambda^q_j \left( \sum_{i \in q} c^\text{unit}_{kij} u^q_{kij} + c^\text{fixed}_{kj} y^q_{kj} \right)
\]

\[
\text{s.t. } \sum_j T \lambda^q_j u^q_{kij} \leq b_{ki} \quad \forall k, i
\]

\[
\sum_k u^q_{kij} = 1 \quad \forall q, j, i \in q
\]

\[
y^q_{kj} \geq u^q_{kij} \geq 0 \quad \forall q, k, j, i \in q
\]

\footnote{Earlier since there was only a single order, we had let \( q \) denote the number of items, which were numbered \( i = 1, \ldots, q \). In this section it is more convenient notationally to let \( q \) instead refer to the set of items in the order.}
(This is identical to the linear program defining $\tilde{J}_{DLP}$ (Jasin and Sinha 2015, p. 1340), except we have let $u_{qkij}^q$ and $y_{qkj}^q$ represent their variables $U_{qkij}^q$ and $Y_{qkj}^q$ divided by $T\lambda_j^q$, respectively.)

In the linear program defining DLP, for any subset $q$ of items ordered from any region $j$, variable $u_{qkij}^q$ represents the proportion of times item $i \in q$ should be fulfilled from FC $k$, with constraint $\sum_k u_{qkij}^q = 1$ for each such item $i$ in the order. Meanwhile, variable $y_{qkj}^q$ represents the probability that a FC $k$ would have to be used at all, which is constrained to be at least $u_{qkij}^q$ for any single item $i \in q$. Note that in an optimal solution we can always assume $y_{qkj}^q = \max_{i \in q} u_{qkij}^q$ for all $q, k, j$. These variables $u_{qkij}^q$ and $y_{qkj}^q$ correspond to our variables $u_{ki}$ and $y_k$ from earlier, where we had dropped scripts $q, j$ to focus on a single multi-item order from a single region.

Moreover, the first constraint enforces that the expected number of times any FC $k$ fulfills any item $i$ (to any region $j$, as part of any subset $q$ containing $i$) does not exceed its starting inventory $b_{ki}$. Finally, the objective value defining DLP represents the total expected cost of the LP benchmark over the time horizon, accounting for unit costs, fixed costs, as well as shortage costs (recalling that there is a null FC $k = 0$). This interpretation of DLP intuitively leads to the following lemma.

**Lemma 5 (Jasin and Sinha (2015)).** For any instance of the problem, the expected cost paid by any dynamic fulfillment policy must be at least the value of DLP for that instance.

**Randomized fulfillment algorithm and reduction result.** In light of the interpretation of the linear program defining DLP above, Jasin and Sinha (2015) also use it to derive the following randomized fulfillment heuristic. First, we solve the LP, hereafter using $u_{qkij}^q, y_{qkj}^q$ to refer to a fixed optimal solution. At each time step $t = 1, \ldots, T$, if an order for subset $q$ comes from region $j$, the heuristic policy randomly chooses an FC $k$ to fulfill each item $i \in q$ according to probabilities $u_{qkij}^q$, independently across time steps, without adapting at all to the remaining inventory. If the chosen FC for an item has stocked out, then that item is simply not fulfilled (i.e. the null FC is used).

This randomized fulfillment heuristic that does not rely on real-time inventory information has been shown to perform well asymptotically, although its theoretical guarantee depends on how exactly FC’s are chosen to fulfill items during each time step, namely, the $\alpha$-competitive rounding
scheme that is used. Jasin and Sinha (2015) show that the unit and shortage costs paid by the randomized fulfillment heuristic is asymptotically optimal relative to the DLP, but the bottleneck is the fixed costs, where every time an order for subset \( q \) comes from region \( j \) (regardless of asymptotics) the cost paid could be \( \alpha \) times as much as the DLP. Here \( \alpha \) depends on \( q \) and \( j \), and using the correlated rounding schemes from Theorems 1 and 2 in this paper in conjunction with the results from Jasin and Sinha (2015) we can always guarantee an \( \alpha \)-competitive rounding scheme where

\[
\alpha = \min \left\{ 1 + \ln(|q|), \left( \min_{i \in q} \max_k u_{ki}^q \right)^{-1}, B(|q|) \right\}
\]

and \( B(\cdot) \) is the function from Jasin and Sinha (2015).

Jasin and Sinha (2015) show that the asymptotic cost paid by the randomized fulfillment heuristic relative to DLP, assuming it chooses the correlated rounding scheme corresponding to the smallest argument in (19) whenever any subset \( q \) is ordered from any region \( j \), is a weighted average of expression (19) over \( q \) and \( j \). To formally state this result, we need to finally define what “asymptotic” means. Here, one considers a scaling regime where for any fixed instance and any \( \theta \geq 0 \), the “scaled instance” is defined to the the one where the horizon length \( T \) has been replaced by \( \theta T \) while each starting inventory \( b_{ki} \) has also been replaced by \( \theta b_{ki} \). Let \( DLP(\theta) \) denote the optimal objective value DLP on the instance scaled by \( \theta \), and let \( ALG(\theta) \) denote the expected cost paid by the randomized fulfillment heuristic on the same scaled instance. The following is then implied by the proof of Theorems 1 and 2 from Jasin and Sinha (2015) (see Jasin and Sinha (2015, p. ec5)), combined with our discussion above.

**Theorem 3.**

\[
\lim_{\theta \to \infty} \frac{ALG(\theta)}{DLP(\theta)} \leq \frac{\sum_{q,k,j} \lambda_j^q c_{kj}^{\text{fixed}} y_{kj}^q \min \left\{ 1 + \ln(|q|), \left( \min_{i \in q} \max_k u_{ki}^q \right)^{-1}, B(|q|) \right\}}{\sum_{q,k,j} \lambda_j^q c_{kj}^{\text{fixed}} y_{kj}^q}. \tag{20}
\]

Since any fulfillment policy must pay cost at least \( DLP(\theta) \) by Lemma 5, this shows that the randomized fulfillment heuristic cannot be worse than the optimal dynamic program by a factor
greater than the RHS of (20). Our guarantee on the RHS of (20) illustrates the power of having different correlated rounding schemes available for different types of orders that could arrive. Jasin and Sinha (2015, Thm. 2) prove the same guarantee with the \( \min \{ \cdot \} \) replaced by just \( B(|q|) \), while Zhao et al. (2020) prove the same result where the upper bound on the RHS is 1 (i.e. prove asymptotic optimality) if there are only two FC’s in the network. We emphasize that all of these asymptotic guarantees that have eliminated the unit and shortage costs only hold if the LP inventory constraints are satisfied in expectation at every time step, justifying why all of these papers study correlated rounding schemes.

6. Numerical Study

We test our \( \alpha \)-competitive rounding schemes on the general multi-item dynamic fulfillment problem formalized in Section 5. We construct instances aimed to model the operations of a large e-tailer in the continental United States, following the setup of Jasin and Sinha (2015) as closely as possible. Our code is in Julia, uses the JuMP (Dunning et al. 2017) package, and is made publicly available at https://github.com/Willmasaur/multi_item_e_commerce_fulfillment.

Regions, fulfillment centers, costs. We allow orders to arrive from regions corresponding to the 99 largest metropolitan areas in the U.S., excluding Honolulu, HI. The arrival rate from each region is scaled by its 2022 population according to the US Cities Database on https://simplemaps.com/data. Meanwhile, we take the 10 largest Amazon.com Inc. fulfillment centers that were operational as of 2015 and assume all items are shipped from one of these centralized FC’s. Following Jasin and Sinha (2015), the fixed cost of packaging a box at any FC \( k \) for any region \( j \) is \( c_{kj}^{\text{fixed}} = 8.759 \), while the cost of shipping a single item \( i \) from any FC \( k \) to any region \( j \) is \( c_{kij}^{\text{unit}} = 0.423 + 0.000541 \text{dist}_{kj} \), where \( \text{dist}_{kj} \) is the air distance between FC \( k \) and region \( j \) in miles.

We note that our city populations and FC locations may differ from Jasin and Sinha (2015), as the exact sources they cited are no longer publicly available. We also procedurally diverge from

\(^2\) Compiled from the information at https://www.mwpvl.com/html/amazon_com.html available with our code.
Jasin and Sinha (2015) by always selecting the largest cities and fulfillment centers, whereas they select randomly when fewer than 99 cities or fewer than 10 FC’s are needed. We believe this to generate a more interesting smaller network, because the 10 largest cities are spread out across the corners while the 5 FC’s are located in the middle, resulting in difficult fulfillment decisions where a city can be “nearby” to multiple FC’s (see the data files provided with our code for details).

**Order types, demand rates, starting inventories.** Order types \( q \) each denote a subset of size up to \( n_{\text{max}} \), from a universe of \( n \) items. For each size in \( 1, \ldots, n_{\text{max}} \), there are \( n_{\text{per}} \) fixed order types, each of which is a subset of the \( n \) items drawn uniformly at random with the correct size. There is also an order type with size 0, which represent the lack of a customer arrival at a time step. Note that the total number of order types is \( 1 + n_{\text{max}} n_{\text{per}} \), which we denote using \( Q \).

The demand probabilities are first split randomly between the order sizes \( 0, 1, \ldots, n_{\text{max}} \), and then for each size, split randomly between the types with that size. This yields a \( Q \)-dimensional probability vector, i.e. a vector whose entries are non-negative and sum to 1. Then, a \( QJ \)-dimensional probability vector is constructed by further splitting each order type among the metropolitan areas according to their populations. This vector \( (\lambda^q_j)_{q,j} \) is then used as input for the dynamic fulfillment problem.

Finally, to determine starting inventories, each FC \( k \) first randomly decides whether to carry each item \( i \), independently with probability \( p_{\text{carry}} \). Then, for each region \( j \), the closest FC \( k \) that carries each item \( i \) is identified as \( \text{closest}_{i,j} \). For an item \( i \), its “demand” at an FC \( k \) is

\[
\text{dem}_{k,i} = \sum_{q \ni i} \sum_j \mathbb{1}(\text{closest}_{i,j} = k) \cdot \lambda^q_j,
\]

where we sum over all queries \( q \) containing a copy of item \( i \), and consider only the regions \( j \) for which FC \( k \) is identified as the closest when summing over arrival probabilities \( \lambda^q_j \). Given these values, starting inventories are then placed so that

\[
b_{ki} = T \text{dem}_{k,i} + z_{\text{safety}} \sqrt{T \text{dem}_{k,i}(1 - \text{dem}_{k,i})}
\]

for all \( k \) and \( i \), where we note that the total demand for item \( i \) closest to FC \( k \) over \( T \) time steps is Binomially distributed, with mean \( T \text{dem}_{k,i} \) and variance \( T \text{dem}_{k,i}(1 - \text{dem}_{k,i}) \). The formula for
$b_{ki}$ is the ideal inventory level to start with according to a Newsvendor model, with safety stock multiplier $z_{\text{safety}}$ set to 0.5 for all items.

We note that our procedures for randomly generating order types, demand rates, and carrying decisions follow  
Jasin and Sinha (2015, EC.3) exactly, in which these methods are justified. The details of these methods can also be found in our code.

**Algorithms.** Like Jasin and Sinha (2015), we test the Myopic fulfillment policy as a baseline, which fulfills each item from the closest FC that carries it, not accounting for split orders and minimizing the number of boxes shipped. We then consider four different algorithms following the randomized fulfillment framework described in Section 5:

- Indep: Independent Rounding, as described in Jasin and Sinha (2015);
- JS: Correlated Rounding scheme of Jasin and Sinha (2015) based on line partitions;
- Dilate: $(1 + \ln(|q|))$-competitive scheme based on dilated opening times (Subsection 2.1);
- ForceOpen: $d$-competitive scheme based on dilated times and forced openings (Subsection 2.2).

Jasin and Sinha (2015) compare Indep and JS to the Myopic fulfillment policy; we additionally compare our new correlated rounding schemes Dilate and ForceOpen.

### 6.1. Experimental Setups

We consider two experimental setups. First, in **Subsection 6.2** we let the number of regions, FC's, items, and time steps be $J = 10, K = 5, n = 20, T = 10^5$ respectively. The queries and starting inventories are generated with $n_{\text{max}}$ varying in $\{2, 5, 10\}$, $n_{\text{per}}$ fixed to 5, and $p_{\text{carry}}$ fixed to 0.75. We note that when $n_{\text{max}} = 5$ this is exactly the “base case” simulated in Jasin and Sinha (2015). We vary $n_{\text{max}}$ to see how different rounding schemes handle different order sizes.

We consider a bigger network in **Subsection 6.3** where the number of regions, FC's, and items are $J = 99, K = 10, n = 100$ respectively. These represent the largest values considered in Jasin and Sinha (2015), and we also increase $T$ to $10^6$ to better capture asymptotic performance. The queries are generated with $n_{\text{max}}$ and $n_{\text{per}}$ increased to 10. Meanwhile, we vary $p_{\text{carry}}$ in $\{0.25, 0.5, 0.75\}$ to investigate how in a big sparse network with a small value $p_{\text{carry}}$, the problem can still be easy and in particular ForceOpen can perform well because $d$ is small.
Table 1 Performance and runtime for the 5 different algorithms under the 3 different values of $n_{\text{max}}$. The best (smallest) average loss in each row is bolded.

| $n_{\text{max}}$ | Myopic | Indep | JS    | Dilate | ForceOpen |
|------------------|--------|-------|-------|--------|-----------|
| $n_{\text{max}} = 2$ Avg. Loss | 4.3%  | 3.1%  | 2.3%  | 2.4%   | 2.4%      |
| $n_{\text{max}} = 5$ Avg. Loss | 12.9% | 14.9% | 9.3%  | 8.3%   | 8.9%      |
| $n_{\text{max}} = 10$ Avg. Loss | 17.7% | 16.5% | 11.7% | 8.6%   | 9.4%      |

| $n_{\text{max}}$ | Runtime per Instance |
|------------------|-----------------------|
| $n_{\text{max}} = 2$ | 0.33s 0.38s 1.17s 0.39s 0.43s |
| $n_{\text{max}} = 5$ | 0.52s 0.59s 3.37s 0.62s 0.72s |
| $n_{\text{max}} = 10$ | 0.84s 1.03s 9.12s 1.09s 1.32s |

For each experimental setup, we randomly generate 30 instances, and then randomly generate 30 arrival sequences for each instance. We use the same arrival sequences for every algorithm to minimize the discrepancy caused by variance in arrival sequences. We fix $z_{\text{safety}}$ to be 0.5 throughout our experiments. All of these aspects match what is done in Jasin and Sinha (2015).

6.2. Performance on Smaller Network with varying Order Size

We consider the first experimental setup with the smaller network, generating 30 random instances for each value of $n_{\text{max}}$ in \{2, 5, 10\}. For each instance, we consider the benchmark DLP described in Section 5 which is a lower bound on the cost of any fulfillment algorithm. We draw 30 arrival sequences to test the performance of the 5 specific algorithms discussed earlier, and compute the average cost of each algorithm over these 30 arrival sequences. We consider how much greater this average cost is than the value of DLP for that instance, expressed as a percentage. The average of these “loss” percentages over the 30 instances are then reported in Table 1 for each algorithm. We also report the average runtime\(^3\) of each algorithm, which we note is the total runtime used to evaluate the 30 arrival sequences for an instance, averaged over instances.

\(^3\) See our code for the exact timing functions used. The time (in seconds) was measured on a Dell Latitude 5510 laptop with an Intel(R) Core(TM) i7-10810U CPU @ 1.10GHz processor and 32GB of RAM.
Observations from results in Table 1. Our algorithms perform favorably in comparison to Myopic, Indep, and JS. Indeed, they pay marginally more cost than JS when \( n_{\text{max}} = 2 \), and overtake JS as soon as \( n_{\text{max}} = 5 \), i.e. orders have sizes between 1 and 5. This is surprising in that the theoretical guarantee of JS is better for the values of \( n \) in this range. Also, we note that the losses of 8.3% and 8.6% for Dilate are relative to an (unreasonable) LP benchmark which does not face any stochastic fluctuation; the loss relative to an actual fulfillment policy that can be implemented (e.g., the optimal dynamic programming policy, given the exponential time required to compute it) would be much smaller. For this reason, we consider the numbers in Table 1 more useful for comparing algorithms than for evaluating absolute performance.

A further, perhaps more salient feature of our algorithms is their simplicity and interpretability. As evidenced in our code, the rounding scheme in Dilate (Algorithm 1) takes 10 lines to write, whereas the rounding scheme in JS took us 100 lines. Also, the average runtime per instance for Dilate is better than JS by a factor of 5–10. This seemingly innocuous difference on the smaller network becomes more pronounced on the bigger network, as we see next.

6.3. Performance on Bigger Network with varying Fulfillment Flexibility

We consider the second experimental setup described in Subsection 6.1. We report average losses and runtimes for each of the 5 algorithms, in the same way as defined in Subsection 6.2. We generate 30 random instances for each value of \( p_{\text{carry}} \in \{0.25, 0.5, 0.75\} \) and report the averages in Table 2.

We note that \( p_{\text{carry}} \) is a measure of fulfillment flexibility, in that a higher value of \( p_{\text{carry}} \) leads to more FC’s being able to fulfill each item and hence more flexibility in the network. Generally this results in a harder fulfillment problem, with a larger value of \( d \), which we recall denotes the maximum number FC’s carrying any item. A lower value of \( p_{\text{carry}} \), on the other hand, results in a smaller \( d \) and a better guarantee for ForceOpen.

Observations from results in Table 2. In this bigger network which also has larger order sizes, all algorithms perform worse. Myopic performs particularly poorly with large order sizes, because it will likely always split the order (since not all FC’s stock all items). We can see a greater
Table 2  Performance and runtime for the 5 different algorithms under the 3 different values of $p_{\text{carry}}$. The best (smallest) average loss in each row is bolded.

| $p_{\text{carry}}$ | Myopic | Indep | JS | Dilate | ForceOpen |
|-------------------|--------|-------|----|--------|-----------|
| 0.25 Avg. Loss    | 34.8%  | 10.2% | 7.6%| 5.6%   | **5.3%**  |
| 0.50 Avg. Loss    | 26.7%  | 23.4% | 17.7%| **12.6%** | 13.1%     |
| 0.75 Avg. Loss    | 22.3%  | 34.2% | 23.2%| **16.1%** | 17.7%     |

$p_{\text{carry}} = 0.25$ Runtime per Instance | 11.23s | 15.43s | 162.31s | 14.24s | 16.88s |
$p_{\text{carry}} = 0.50$ Runtime per Instance | 11.89s | 18.25s | 162s    | 17.01s | 19.25s |
$p_{\text{carry}} = 0.75$ Runtime per Instance | 13.01s | 19.33s | 169.27s | 18.59s | 22.09s |

separation between the performance of our algorithms, Dilate and ForceOpen, vs. the performance of the other algorithms. And while we had always observed ForceOpen to both be more complex and perform slightly worse than Dilate, we now see that when $p_{\text{carry}} = 0.25$, it in fact performs better. This is related to its theoretical guarantee—the value of $d$ tends to be smaller when $p_{\text{carry}} = 0.25$, because each item in expectation is carried in only 2.5 FC’s.

There is also now a factor-10 speedup in the runtime of our algorithms compared to JS, which means that the time to finish per instance is on the order of tens of seconds instead of minutes.

**Takeaways from numerical study.** Under the randomized fulfillment framework of Jasin and Sinha (2015), one should generally default to Dilate to perform correlated rounding, because it is simple to implement, fast to run, and performs either the best or close to the best across the different setups. For orders with 2 items, JS may perform slightly better. In large sparse networks where each item is carried at very few FC’s, ForceOpen may perform slightly better.

7. Conclusion

We provide the first improvements to the celebrated correlated rounding procedure of Jasin and Sinha (2015) for the problem of multi-item e-commerce order fulfillment. We derive rounding schemes with guarantees of $1 + \ln(q)$ and $d$ respectively, where $q$ is the number of items in the order and $d$ is the maximum number of fulfillment centers containing any item. The first
of these guarantees improves the guarantee of $\approx q/4$ from Jasin and Sinha (2015) by an order of magnitude, in terms of the dependence on $q$. We also show both of our guarantees to be tight, by deriving new relationships with the Set Cover problem. Testing under a realistic setup originated by Jasin and Sinha (2015), we find the improvement provided by our new rounding schemes to in fact be greater than what their theoretical guarantees suggest.

**Acknowledgments**

This research is partially funded by a grant from Amazon.com Inc., which is awarded through collaboration with the Columbia Center of AI Technology (CAIT). An earlier version of this paper was submitted and accepted to the Manufacturing & Service Operations Management (MSOM) conference Special Interest Group (SIG) for Supply Chain Management, 2022, whose anonymous reviewers provided excellent comments. The author also thanks Levi DeValve, Stefanus Jasin, Aravind Srinivasan, Yehua Wei, and Linwei Xin for sharing background about this problem.

**References**

Acimovic J, Farias VF (2019) The fulfillment-optimization problem. *Operations Research & Management Science in the age of analytics*, 218–237 (INFORMS).

Acimovic J, Graves SC (2015) Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing & Service Operations Management* 17(1):34–51.

Buchbinder N, Naor JS, et al. (2009) The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science* 3(2–3):93–263.

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

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

Jasin S, Kumar S (2012) A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* 37(2):313–345.

Jasin S, Sinha A (2015) An lp-based correlated rounding scheme for multi-item ecommerce order fulfillment. *Operations Research* 63(6):1336–1351.
Lei Y, Jasin S, Sinha A (2018) Joint dynamic pricing and order fulfillment for e-commerce retailers. *Manufacturing & Service Operations Management* 20(2):269–284.

Lei Y, Jasin S, Uichanco J, Vakhutinsky A (2021) Joint product framing (display, ranking, pricing) and order fulfillment under the multinomial logit model for e-commerce retailers. *Manufacturing & Service Operations Management*. 

Motwani R, Raghavan P (1995) *Randomized algorithms* (Cambridge university press).

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

Vazirani VV (2001) *Approximation algorithms*, volume 1 (Springer).

Wei L, Kapuscinski R, Jasin S (2021) Shipping consolidation across two warehouses with delivery deadline and expedited options for e-commerce and omni-channel retailers. *Manufacturing & Service Operations Management* 23(6):1634–1650.

Xu PJ, Allgor R, Graves SC (2009) Benefits of reevaluating real-time order fulfillment decisions. *Manufacturing & Service Operations Management* 11(2):340–355.

Zhao Y, Wang X, Xin L (2020) Multi-item online order fulfillment: A competitive analysis. *Chicago Booth Research Paper* (20-41).