Online Rounding Schemes for k-Rental Problems

Hossein Nekouyan*   Bo Sun†   Raouf Boutaba‡   Xiaoqi Tan§

July 30, 2025

Abstract

We study two online resource allocation problems with reusability in an adversarial setting, namely kRental-Fixed and kRental-Variable. In both problems, a decision-maker manages k identical reusable units and faces a sequence of rental requests over time. We develop theoretically grounded relax-and-round algorithms with provable competitive ratio guarantees for both settings. For kRental-Fixed, we present an optimal randomized algorithm that achieves the best possible competitive ratio. The algorithm first computes an optimal fractional allocation using a price-based approach, and then applies a novel lossless online rounding scheme to obtain an integral solution. For kRental-Variable, we first establish the impossibility of achieving lossless online rounding. We then introduce a limited-correlation rounding technique that treats each unit independently while introducing controlled dependencies across allocation decisions involving the same unit. Combined with a carefully-crafted price-based method for computing the fractional allocation, this approach yields an order-optimal competitive ratio for the variable-duration setting.

1 Introduction

This paper investigates adversarial online allocation problems involving *reusable* resources, which we refer to as *k-rental* problems. In the k-rental setting, a seller (or decision-maker) manages an inventory of k identical units of a resource, say, k balls. A sequence of requests arrives over time, each seeking to *rent* one ball for a specified duration. Once allocated, a ball becomes unavailable until the rental period ends, after which it returns to the inventory. Each request also specifies a unique valuation for renting a ball. The objective is to maximize the total valuation of all accepted requests.

In the special case without reusability (i.e., the rental durations are infinite), numerous variants of k-rental problems have been extensively studied under different arrival models. These include the secretary problem in the random-order model [13], prophet inequalities in stochastic IID or non-IID settings [3, 11, 22], and different forms of adversarial online selection problems, such as k-search and one-way trading [7, 26, 23, 32], the online knapsack problem [30, 31, 33], and their applications in revenue management [28, 2].

In this paper, we study two adversarial k-rental problems, denoted kRental-Fixed and kRental-Variable. In kRental-Fixed, all requests have a fixed and identical rental duration, but their valuations are uncertain. The kRental-Variable setting generalizes this by allowing rental durations to vary across requests, with each request's valuation assumed to be linear in its duration.

---

*University of Alberta. Email: nekouyan@ualberta.ca

†University of Ottawa. Email: bo.sun@uottawa.ca

‡University of Waterloo. Email: rboutaba@uwaterloo.ca

§University of Alberta. Email: xiaoqi.tan@ualberta.ca

For the kRental-Fixed problem, [4] considers a more general online bipartite matching model in which each matched offline node becomes available again after a fixed d time units. Their algorithm achieves a competitive ratio of 0.589, though it does not address the optimality of their algorithm or lower bounds of the kRental-Fixed problem. They also proposed a second algorithm using a correlation-based rounding scheme that achieves a 0.505-competitive ratio which is very close to the 0.5 ratio attained by the greedy algorithm in this setting. However, the primary limitation of their approach lies in the rounding step, which lacks sufficient effectiveness to deliver tight performance guarantees. A more recent work by [6] examines an online job assignment problem that overlaps with both kRental-Fixed and kRental-Variable. They develop *deterministic* algorithms that achieve order-optimal competitive ratios, but only in the *large-inventory* regime where $k \to \infty$. This leaves open the important question of designing *randomized* algorithms with tight guarantees for both kRental-Fixed and kRental-Variable across all ranges of inventory sizes, including small and moderate regimes (i.e., k is finite).

1.1 Our Contributions

We address the above open question affirmatively by developing randomized algorithms for k-rental under general inventory settings. Our algorithms achieve the optimal competitive ratio for kRental-Fixed and a near-optimal performance guarantee for kRental-Variable. Specifically, our contributions are twofold:

γ-OCR and lossless online rounding schemes.   We introduce an online rounding subroutine termed γ-Online Correlated k-Rental (γ-OCR) that captures the core challenge of rounding any feasible online fractional allocations in reusable settings with fixed rental durations. The goal is to round each fractional allocation to an integral one, allocating a unit with probability at least equal to its fractional value multiplied by $\gamma \in [0, 1]$. As a warm-up, we present an independent rounding scheme and show that it is lossless when the inventory is large but fails in small-inventory cases. To address this, we develop a new online rounding algorithm, 1-OCR, which introduces correlations across time steps to achieve lossless rounding. We believe this method has broader applicability and may be of independent interest.

Implications and insights of γ-OCR for pricing reusable resources.   Our online rounding framework γ-OCR yields both positive and negative results for the kRental-Fixed and kRental-Variable problems, respectively. For kRental-Fixed, we design an optimal randomized algorithm grounded in the 1-OCR framework. This algorithm first computes an optimal online fractional solution using a price-based approach and then applies the 1-OCR rounding scheme to obtain an integral allocation. In contrast, for kRental-Variable with variable rental durations, we establish a fundamental impossibility result: no online rounding scheme can convert an arbitrary fractional solution into an integral one while preserving the same competitive ratio. To address this challenge, we propose a randomized algorithm based on a limited-correlation rounding scheme that treats different units independently while introducing controlled dependencies across allocation decisions involving the same unit. Although the rounding step is inherently lossy, we prove that the resulting algorithm attains an order-optimal, best-known performance guarantee for kRental-Variable.

Techniques.   Our overarching approach follows the *relax-and-round* framework, which first computes a fractional allocation for a relaxed version of the problem using a price-based method, and then rounds this solution to obtain an integral allocation. The second step, i.e., the rounding procedure, is more nuanced and central to our contribution. From a technical standpoint, to derive a pricing function that produces the optimal fractional solution for kRental-Fixed, we adopt an online primal-dual approach to guide the design of an appropriate pricing function. For kRental-Variable, we employ the LP-free certificate framework developed in [16] to construct the pricing function. In this approach, designing an α-competitive

2

algorithm reduces to finding a feasible solution to a system of *delayed differential inequalities*, which the pricing function of DOP-ϕ-VARIABLE must satisfy in order to achieve the desired competitiveness. This system, parameterized by the achievable competitive ratio, characterizes the design of a pricing function that attains the smallest possible competitive ratio.

1.2   Related Work

Online resource allocation—the process of assigning limited resources to sequentially arriving requests to maximize social welfare or profit—is a central topic in computer science and operations research. In the following we briefly mention important papers from the literature related to this work.

Online scheduling.   The adversarial online interval scheduling problem, introduced by [25], considers the task of scheduling a sequence of intervals (jobs) on a single server as they arrive in order of their start times. Assuming that the minimum and maximum job durations are unknown, their randomized algorithm based on a converging sequence of coin flip probabilities achieves an $O\big((\ln \Delta)^{1+\varepsilon}\big)$ competitive ratio for any $\varepsilon > 0$, where Δ is the ratio of the longest to the shortest job duration. Motivated by this work, several subsequent studies (e.g., [9, 17, 15]) have explored various extensions of the problem. For instance, [15] examined a variant in which the minimum and maximum job durations are known, referred to as the *online reservation problem*, and proposed a static pricing algorithm with a performance guarantee of $3 \cdot \big(1 + \ln(\Delta)\big)$. However, the static threshold approach may be ill-suited for online allocation involving reusable resources, where the algorithm must manage more complex dynamics and schedule jobs to maximize total resource utilization. In Section 4, we investigate whether a dynamic pricing scheme, one that continuously updates resource prices in response to changing market conditions, can achieve stronger performance guarantees. Additionally, we develop an online rounding scheme that, when integrated with the dynamic pricing mechanism, effectively converts fractional solutions into integral ones while maintaining robust performance guarantees.

Online matching with reusable resources.   Several studies, including [4, 30, 14, 20, 6], have investigated the online matching problem and its variants in settings with resource reusability. In particular, as previously discussed, [4] examined an online bipartite matching problem in which resources are reusable. This setting extends the classical online matching model introduced by [24] by allowing each matched offline node to become available again after d time units. In Section 3, we focus on a special case of this model involving a single resource type and valuation uncertainty. We explore whether improved competitive ratios and stronger correlation schemes can be achieved compared to their results. [20] studied adversarial reusability in the context of online assortment planning, where the kRental-Variable problem can be reduced to their model. Their work also centers on deterministic algorithm design. Within the kRental-Variable framework, their algorithm achieves a competitive ratio of $4 \ln\big(\frac{d_{\max}}{d_{\min}}\big)$ as the inventory size tends to infinity, where $d_{\max}$ and $d_{\min}$ denote the maximum and minimum rental durations, respectively. More recently, [6] considered a generalized version of the kRental-Variable problem within the online matching framework. Their model incorporates uncertainty in each request's per-unit-time valuation over the requested interval and across different item types. Given the close connection between their formulation and ours, we provide a detailed comparison in Appendix J, deferred due to space constraints.

Online rounding.   Recent work in computer science and operations research has demonstrated the effectiveness of online rounding frameworks [8, 27], which typically adopt a *relax-and-round* paradigm. Notable progress has been made in designing innovative online rounding schemes in recent years. For

example, [8] introduced a subroutine called Online Correlated Selection, which imposes negative correlation across selected pairs to achieve improved competitive ratios over classical greedy algorithms, which are limited to a $\frac{1}{2}$-competitive ratio. Building on this idea, [4] developed the Online Correlated Rental method for settings with reusable resources and fixed rental durations. Similar correlation-based techniques have also been adapted to the Adwords problem by [19]. In the context of stochastic optimization, several studies (e.g., [10, 1]) have proposed related rounding techniques, such as online contention resolution schemes, demonstrating their effectiveness in applications including prophet inequalities and online stochastic matching.

2 A Lossless Rounding Scheme for Online Correlated k-Rental

We first introduce Online Correlated k-Rental (OCR), which is an online rounding subroutine for k-rental problems. The OCR subroutine captures the key challenge of rounding a fractional solution in k-rental problems where resources are reusable, and
serves as a building block for algorithm design in Section 3.

2.1 γ-OCR: Definitions, Objectives, and Challenges

Definition 1 (γ-OCR). *Consider a set of k identical balls, each uniquely labeled from the set $\{1, 2, \ldots, k\}$. Each ball can be rented to a player for a fixed duration of d time units, after which it becomes available for reuse. A sequence of N players arrives one by one, where each player n is characterized by a tuple $(\hat{x}_n, a_n)$. Here, $\hat{x}_n \in [0, 1]$ denotes the target probability with which the procedure should assign a ball to player n, and a_n is the arrival time of player n. For a fixed $\gamma \in [0, 1]$, a γ-OCR is an online rounding scheme that guarantees renting a ball to each player $n \in [N]$ with probability at least $\gamma \hat{x}_n$, for all input instances $\{(\hat{x}_n, a_n)\}_{n \in [N]}$ satisfying the following condition:*

$$\hat{x}_n \leq \min \left\{ 1, \ k - \sum_{j \in [n-1]} \hat{x}_j \cdot \mathbb{I}_{\{a_j + d > a_n\}} \right\}, \qquad \forall n \in [N]. \tag{1}$$

As a rounding scheme, γ-OCR is a randomized online algorithm that makes irrevocable decisions upon the arrival of each player, either assigning a ball to the player or rejecting it. γ-OCR focuses on the inputs that satisfy the regularity conditions in Eq. (1). In particular, the term $\sum_{j \in [n-1]} \hat{x}_j \cdot \mathbb{I}_{\{a_j + d > a_n\}}$ on the right-hand side quantifies the cumulative targeted probabilities of players who arrived prior to player n and whose rental intervals (each of length d) overlap with that of player n. This sum accounts for the expected occupancy of the balls at the time of player n's arrival. Furthermore, the condition in Eq. (1) also guarantees that $\hat{x}_n \leq 1$ for all $n \in [N]$, thereby ensuring that the assignment probability for any individual player does not exceed the availability of a single unit. Violation of this constraint would imply that the total targeted assignment at the arrival of player n exceeds the inventory limit of k units.

The goal is to design a γ-OCR that maximizes γ. A 1-OCR is referred to as a *lossless* rounding scheme, as it guarantees renting a ball to each player $n \in [N]$ with probability at least $\hat{x}_n$. Designing such a scheme, however, is non-trivial. As a warm-up, we first present a simple independent rounding algorithm that incurs rounding losses. This scheme highlights the limitations of uncorrelated rounding and underscores the need to correlate the allocation decision for each newly arriving player with the decisions made for previously served players who still hold rented balls.
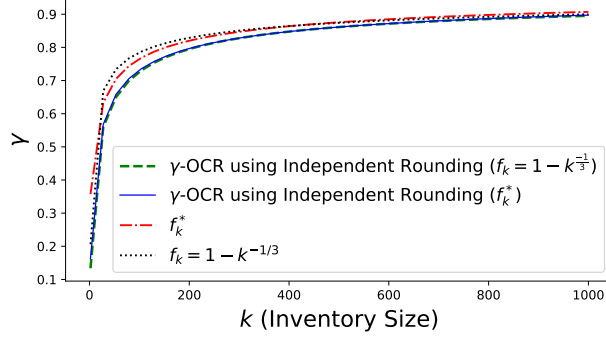
Figure 1: Comparison of the approximate parameter $f_k = 1 - \frac{1}{k^{1/3}}$ versus the optimal parameter f_k^*, and the resulting γ values for Algorithm 1 using f_k and f_k^*.

2.2 Warm Up: A Simple Independent Rounding Scheme

Algorithm 1: γ-OCR using Independent Rounding

---

1 Input: Number of balls k, rental duration d, parameter $f_k \in [0,1]$.

2 Output: Assignment $z_n \in \{0,1\}$ for each player $n \in [N]$.

3 for *each player n with the tuple $(a_n, \hat{x}_n)$* **do**

4   Compute the number of balls currently rented at time a_n: $y_n = \sum_{j=[n-1]} z_j \cdot \mathbb{I}_{\{a_j+d>a_n\}}$.

5   Sample a random seed $S_n \sim U(0,1)$.

6   **if** $S_n \le f_k \cdot \hat{x}_n$ **and** $y_n < k$ **then**

7 $z_n \leftarrow 1$;   ▷ Allocate the $(y_n + 1)$-th available ball to player n

8   **end**

9   **else**

10 $z_n \leftarrow 0$.   ▷ Reject request n

11   **end**

12 end

Consider the rounding scheme in Algorithm 1, which is motivated by rounding schemes in the approximation algorithms literature.

It samples a random seed S_n independently at the arrival of each player n from the uniform distribution $U(0,1)$. A ball is then allocated to player n if $S_n \le f_k \cdot \hat{x}_n$ and at least one ball is available. The parameter $f_k \in [0,1]$ serves as a downscaling factor that depends on the total number of balls, k, and targets to reduce the allocation probability for each player. This downshift increases the likelihood that a ball will be available for future arrivals, allowing more players to be served.

Proposition 1. *Given $f_k \in [0,1]$, Algorithm 1 is γ_k-OCR, where $\gamma_k = f_k \cdot \left(1 - \exp\left(-\frac{(k-f_k \cdot k)^2}{f_k \cdot k + k}\right)\right)$.*

The proof of Proposition 1 is based on the Chernoff bound inequality and is provided in Appendix A. This result also informs the design of the downscaling parameter f_k as a function of k to maximize γ_k. The optimal value f_k^* cannot be obtained in closed form, since it involves solving a transcendental equation of the form $A(x) = \ln(1 - D(x))$. However, it can be efficiently computed using numerical methods such as binary search, and is illustrated in Figure 1. Based on the numerical results, we also observe that the optimal parameter f_k^* closely follows a simple expression $f_k = 1 - \frac{1}{k^{1/3}}$.

Figure 1 shows that the performances of Algorithm 1 with $f_k = 1 - \frac{1}{k^{1/3}}$ and f_k^* are also close. In addition, as seen in the figure, for large values of k the resulting γ approaches one; however, for small inventory

5

sizes (small values of k), the algorithm is not an effective rounding scheme. This motivates our design for a lossless rounding scheme.

2.3 A 1-OCR Through Dependent Rounding

In this subsection, we propose a 1-OCR by correlating the allocation decisions in Algorithm 2.

Key idea of creating negative correlation. At a high level, the key idea of Algorithm 2 is to correlate each current allocation decision with past decisions involving players whose allocated balls are expected to return sooner. Specifically, prior to the arrival of any player, the algorithm samples a single random seed $r \sim U(0, 1)$, which remains fixed throughout the entire execution and serves as the only source of randomness. This shared seed induces correlation across allocation decisions for different players. For each player n, the algorithm uses this seed to determine whether to allocate a ball. It maintains two pointers: $m_n \in \{1, 2, \ldots, k\}$ and $p_n \in [0, 1]$. The first pointer, m_n, indicates the ball under consideration for allocation to player n; this ball may be unavailable if it is currently assigned to a previous player. The second pointer, p_n, specifies a subinterval of $[0, 1]$ used to guide the allocation decision. For each player n, the algorithm proceeds based on the relationship between p_n and $\hat{x}_n$, as described in the following two cases.

- **Case 1:** If $p_n + \hat{x}_n < 1$, then the algorithm allocates ball m_n to player n if $r \in [p_n, p_n + \hat{x}_n)$ and the ball is available in the system. Otherwise, the player is rejected and no ball is allocated.

- **Case 2:** If $p_n + \hat{x}_n \geq 1$, then the algorithm allocates ball m_n if $r \in [p_n, 1]$ (and the ball is available in the system), or allocates ball $m_n + 1$ if $r \in [0, p_n + \hat{x}_n - 1)$. Since $\hat{x}_n \leq 1$, these intervals are non-overlapping, ensuring that no more than one ball is allocated.

Key invariant of Algorithm 2. It is worth noting that Algorithm 2 does not verify the availability of balls m_n and $m_n + 1$ before assigning them to players. This omission is justified by some key invariants that we establish in Proposition 2 below concerning the availability of these balls.

Proposition 2 (Invariants of Algorithm 2). *Upon the arrival of the n-th player, the following holds:*

- *If $p_n + \hat{x}_n < 1$ and the random seed $r \in [p_n, p_n + \hat{x}_n)$, then the ball m_n is available.*

- *If $p_n + \hat{x}_n \geq 1$ and the random seed $r \in [p_n, 1]$, then the ball m_n is available.*

- *If $p_n + \hat{x}_n \geq 1$ and the random seed $r \in [0, p_n + \hat{x}_n - 1)$, then the ball $m_n + 1$ is available.*

The proof of the above proposition is provided in Appendix B.1. The pointer mechanism described therein plays a critical role in synchronizing the scheduling of different balls for consecutive players. It also maintains a record of subintervals within $[0, 1]$ such that, if the random seed r falls within these intervals, either ball m_n or m_{n+1} remains available. As explained in the context of Algorithm 2, the total length of the subinterval used for decision-making for each player n is precisely $\hat{x}_n$. Since r is sampled from the uniform distribution $U(0, 1)$ and a ball is assigned when r lies within the corresponding subinterval, each player receives a ball with the target probability $\hat{x}_n$.

Leveraging the key invariants established in Proposition 2, we can conclude that Algorithm 2 is lossless.

Theorem 1. *Algorithm 2 is a 1-OCR, namely, it is a lossless rounding scheme.*

Due to the space limit, we defer the full proof of the above theorem to Appendix B.2, and provide a detailed example to explain how Algorithm 2 works in Appendix B.3.

Algorithm 2: 1-OCR using Dependent Rounding

---

1 **Input**: Number of balls k, rental duration d.

2 **Output**: Assignment $z_n \in \{0, 1\}$ for each player $n \in [N]$.

3 **Initialize:** Set $m_1 = 1$, $p_1 = 0$. sample a random seed $r \sim U(0, 1)$.

4 **for** *each request n with the tuple $(a_n, \hat{x}_n)$* **do**

5   **if** $\sum_{j \in [n]} \hat{x}_j \cdot \mathbb{I}_{\{a_j + d > a_n\}} > k$ **then**

6        $z_n \leftarrow 0$.   ▷ Reject player n

7   **end**

8   **if** $p_n + \hat{x}_n < 1$ **then**

9   **if** $r \in [p_n, p_n + \hat{x}_n)$ **then**

10           $z_n \leftarrow 1$.   ▷ Assign ball m_n to player n

11   **end**

12   **else**

13           $z_n \leftarrow 0$.   ▷ Reject player n

14   **end**

15   Update $p_{n+1} = p_n + \hat{x}_n$ and $m_{n+1} = m_n$.

16   **end**

17   **else**

18   **if** $r \in [p_n, 1]$ **then**

19           $z_n \leftarrow 1$.   ▷ Allocate ball m_n to player n

20   **end**

21   **else if** $r \in [0, \hat{x}_n + p_n - 1)$ **then**

22           $z_n \leftarrow 1$.   ▷ Allocate ball $m_n + 1$ to player n

23   **end**

24   **else**

25           $z_n \leftarrow 0$.   ▷ Reject player n

26   **end**

27   Update $p_{n+1} = \hat{x}_n + p_n - 1$ and $m_{n+1} = m_n + 1$ (if $m_{n+1} > k$, then set $m_{n+1} = 1$).

28   **end**

29 **end**

---

3   kRental-Fixed: k-Rental with Fixed Rental Durations

In this section, we define the kRental-Fixed problem and present an optimal randomized algorithm based on the rounding scheme, 1-OCR, developed in the previous section.

3.1   Problem Formulation and Assumptions

Let us formally define the online k-rental problem (kRental-Fixed) as follows. A decision maker has k units of an item and allocates them to online arriving requests. Each request $n \in [N]$ asks to rent one item for d time units. The n-th request arrives at time a_n, and has a valuation v_n for renting one item. In particular, if an item is allocated to request n, the item is rented starting from a_n, and becomes available again at time $a_n + d$.

Let $x_n \in \{0, 1\}$ denote the decision to accept or reject the n-th request. The objective of the problem is to maximize the total valuation of requests that receive an item allocation, i.e., $\sum_{n \in [N]} x_n \cdot v_n$, while respecting the constraints on available items.

Let $I = \{v_n, a_n\}_{n \in [N]}$ denote an instance of kRental-Fixed. The maximum valuation from the optimal clairvoyant algorithm, $\mathsf{OPT}(I)$, can be determined by solving the following optimization problem:

$$\max_{x_n} \quad \sum_{n \in [N]} v_n \cdot x_n, \tag{2a}$$

$$\text{s.t.} \quad \sum_{j \in [n]} x_j \cdot \mathbb{I}_{\{d + a_j > a_n\}} \leq k, \quad \forall n \in [N], \tag{2b}$$

$$x_n \in \{0, 1\}, \quad \forall n \in [N]. \tag{2c}$$

The constraint (2b) ensures that at any point in time throughout the horizon, no more than k items are allocated. The left-hand-side of this constraint counts the number of units of the item allocated at that moment, including the decision made for the arriving request n. Thus, it is sufficient to enforce this constraint only at the arrival of each request n, when the available inventory may decrease.

In the online setting, the decision maker must make an irrevocable decision to accept or reject each request upon its arrival. The uncertainty regarding future requests' valuations and the overall demand for the items makes this decision challenging. To achieve a bounded performance, we follow the literature and assume the valuations of requests are bounded.

Assumption 1. *All valuations are within the range* $[v_{\min}, v_{\max}]$, *i.e.,* $v_n \in [v_{\min}, v_{\max}], \forall n \in [N]$.

Let $\mathcal{I}$ denote the set of all instances of the kRental-Fixed that satisfy Assumption 1. Our goal is to design online algorithms whose objective is competitive with that of $\mathsf{OPT}(I)$ for every instance $I \in \mathcal{I}$. Specifically, an online algorithm ALG is said to be α-competitive if, for any input instance I, the following inequality holds $\alpha \geq \frac{\mathsf{OPT}(I)}{\mathbb{E}[\mathsf{ALG}(I)]}$, where the expectation $\mathbb{E}[\mathsf{ALG}(I)]$ is taken over the randomness of the online algorithm.

In the following section, we present an algorithm for the kRental-Fixed problem that achieves the minimum possible competitive ratio among all online algorithms. This result strictly improves upon the deterministic algorithm proposed by [6], which attains only an order-optimal competitive ratio in the asymptotic regime where the inventory size tends to infinity.

3.2 DOP-ϕ-FIXED: A Relax-and-Round Algorithm for kRental-Fixed

We introduce a randomized online algorithm, DOP-ϕ-FIXED, presented in Algorithm 3, which is based on a general *relax-and-round* framework. In the *relaxation step*, we design an online algorithm for a continuous version of problem (2) by relaxing the integrality constraint to $x_n \in [0, 1]$ for all $n \in [N]$. This algorithm produces an online fractional solution $\{\hat{x}_n\}_{n \in [N]}$. In the *rounding step*, the algorithm converts the fractional solution $\hat{x}_n$ into an integral decision $x_n \in \{0, 1\}$.

Core idea for obtaining fractional allocation: Pseudo-utility maximization in Eq. (3). Upon the arrival of request n, the algorithm first computes an expected utilization level $y_n = \sum_{j=1}^{n-1} x_j \cdot \mathbb{I}_{\{a_j + d > a_n\}}$, which is the expected number of resource units that are currently rented by the previous requests whose rental durations overlap with that of request n. Using this expected utilization level, the algorithm determines the fractional allocation by solving a *pseudo-utility maximization* problem as described in Eq. (3). The first term $x \cdot v_n$ is the valuation from request n, and the second term is a *pseudo-cost* of renting x unit of an item when the current utilization level is y_n. Specifically, the pseudo-cost, $k \int_{y_n/k}^{(y_n+x)/k} \phi(\eta) \, d\eta$, is estimated using a carefully-designed, normalized *marginal pricing* function $\phi : [0, 1] \rightarrow [v_{\min}, v_{\max}]$. Thus, the integration of ϕ over the resource utilization interval $[\frac{y_n}{k}, \frac{y_n+x}{k}]$ gives the pseudo-cost.

In this context, the fractional allocation $\hat{x}_n$ represents the optimal fraction of a resource unit to allocate to request n, given their valuation v_n and the pricing rule defined by ϕ.

Rounding subroutine: 1-OCR. In this step, the fractional solution $\hat{x}_n$ is passed to the lossless online rounding procedure 1-OCR. This procedure generates an integral decision x_n on whether to accept the request. This procedure operates as an online algorithm, with an instance of this algorithm initiated at the start of Algorithm 3. As each request n arrives at time a_n, the 1-OCR procedure receives a probability value $\hat{x}_n \in [0, 1]$ (generated by the relaxation step) as input at time a_n. Then it makes an integral decision x_n on whether to accept or reject the request. All rounding decisions are based on one random seed, which is fixed when Algorithm 3 initiates the 1-OCR instance and remains the same for all requests.

Algorithm 3: Duration-Oblivious Price-based Algorithm with Lossless Rounding for kRental-Fixed (DOP-ϕ-FIXED);

---

1 **Input:** Pricing function $\phi : [0, 1] \rightarrow [v_{\min}, v_{\max}]$.
2 **Initiate** an instance of 1-OCR procedure.
3 **for** *each arriving request n with (v_n, a_n)* **do**
4 $\quad$ Compute the expected utilization level at time a_n: $y_n = \sum_{j=1}^{n-1} x_j \cdot \mathbb{I}_{\{a_j+d>a_n\}}$.
5 $\quad$ **if** $y_n < k$ **then**
6 $\quad\quad$ Compute the fractional allocation: $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Relax step

$$\hat{x}_n = \underset{x \in [0, \min\{1, k-y_n\}]}{\arg\max} \; x \cdot v_n - k \int_{\eta=y_n/k}^{(y_n+x)/k} \phi(\eta) \, d\eta. \qquad (3)$$

7 $\quad\quad$ Decide the integral allocation by Algorithm 2: $x_n = $ 1-OCR $(\hat{x}_n, a_n)$. $\quad$ ▷ Round step
8 $\quad$ **end**
9 $\quad$ **else**
10 $\quad\quad$ $x_n = 0$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Reject request n
11 $\quad$ **end**
12 **end**

I

Before leaving this subsection, we remark that the pseudo-utility maximization that yields the fractional allocation in Eq. (3) differs fundamentally from the pseudo-cost frameworks of [6, 30]. Those works adopt a *forward-looking* approach that estimates the cost of accepting a request using the *entire* utilization trajectory over the request's rental interval. Thus, the pseudo-cost is a functional of the utilization curve throughout the interval, not merely the level observed at arrival. In contrast, our algorithm adopts a much simpler *myopic* approach that computes its pseudo-cost just based on the *current* utilization at the moment the request arrives (hence the term "*duration-oblivious*"). We next show that it is a simpler yet still competitive pricing rule.

3.3 Theoretical Guarantee of DOP-ϕ-FIXED

As explained in Section 3.2, DOP-ϕ-FIXED employs the 1-OCR procedure to round the fractional allocation $\hat{x}_n$ at each time step and to make an integral decision. For this procedure to function correctly, the fractional solution generated by the relax step must satisfy the feasibility conditions in Eq. (1) for 1-OCR.

Lemma 1 (Feasibility of Fractional Solutions). *In DOP-ϕ-FIXED, the input of the round step (line 7 of Algorithm 3), $\{(\hat{x}_n, a_n)\}_{n \in [N]}$, satisfies the feasibility condition in Eq. (1).*

The above lemma naturally follows from the design of DOP-ϕ-FIXED. Based on this, we prove the performance guarantee for DOP-ϕ-FIXED.

Theorem 2. *DOP-ϕ-FIXED is $\left(1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)\right)$-competitive for kRental-Fixed when ϕ is given by*

$$\phi(y) = v_{\min} \cdot \exp\left(\left(1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)\right) \cdot y - 1\right), \quad y \in [0, 1]. \tag{4}$$

The complete proof of Theorem 2 is provided in Appendix C. Our proof follows the well-established online primal-dual (OPD) framework, with a notable deviation from the conventional approach: the updates to the dual variables are deferred until after the arrival of the final request similar to the work done in [20]. Specifically, the dual linear program involves variables $\{u_n, \lambda_n\}_{n \in [N]}$. In line with the interpretation in [5], λ_n can be viewed as the per-unit price of the item at the arrival of request n, while u_n represents the utility that request n derives under the pricing scheme that determines the fractional allocation. A key step in the OPD method is the update of the variable λ_n. For each request n, it suffices to increase the price only at the arrival time of request v_n^*, defined as the *latest-arriving request whose interval overlaps with that of request n*. This suffices because any future request whose allocation might be influenced by that of request n must also have an interval overlapping with v_n^*. Therefore, updating the item price at the arrival of v_n^* ensures the the dual constraint getting satisfied.

Finally, we can show that DOP-ϕ-FIXED achieves the optimal competitive ratio by showing a matching lower bound as follows.

Proposition 3 (Lower Bound). *Under Assumption 1, no online algorithm, deterministic or randomized, can obtain a competitive ratio better than $1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$ for the kRental-Fixed problem.*

The proof of the above proposition is presented in Appendix D. Following the approach of [29, 21], we construct a *family of hard instances* to derive the desired lower bound. The instance begins with k requests, each with valuation $v_{\min}$. These are followed by successive batches of requests with monotonically increasing valuations, eventually reaching $v_{\max}$. All requests arrive within an arbitrarily small time window that is negligible relative to the deadline d. By applying the representative function-based approach [29, 21] to this family of instances, we derive the optimal online algorithm and establish a matching lower bound applicable to all online algorithms.

4 kRental-Variable: k-Rental with Variable Rental Durations

This section focuses on the kRental-Variable problem, which generalizes the kRental-Fixed problem studied in Section 3 by allowing variable rental durations.

4.1 Problem Statement and Assumptions

We formally define the k-rental problem with variable rental durations (kRental-Variable) as follows. A decision maker has k identical units of a resource to allocate to N requests arriving online. Each request $n \in [N]$ arrives at time a_n and asks to rent one unit of the resource for d_n time units, where the rental duration d_n may vary across requests. If a unit is allocated to request n, it remains occupied until time $a_n + d_n$, after which it becomes available again for future requests. Each request has a valuation equal to its rental duration d_n. Let $x_n \in \{0, 1\}$ indicate whether request n is accepted ($x_n = 1$) or rejected ($x_n = 0$). The objective is to maximize the total valuation of accepted requests, $\sum_{n \in [N]} x_n d_n$, subject to the resource constraint corresponding to the finite inventory capacity.

Let $I = \{a_n, d_n\}_{n \in [N]}$ denote an instance of the problem. The performance of the optimal clairvoyant

algorithm on instance I, $\mathsf{OPT}(I)$, can be computed based on following integer linear program:

$$\max_{x_n} \quad \sum\nolimits_{n \in [N]} x_n \cdot d_n, \tag{5a}$$

$$s.t. \quad \sum\nolimits_{j \in [n]} x_j \cdot \mathbb{I}_{\{d_j + a_j > a_n\}} \le k, \quad \forall n \in [N], \tag{5b}$$

$$x_n \in \{0, 1\}, \quad \forall n \in [N]. \tag{5c}$$

To achieve a bounded competitive ratio, we still impose constraints on the adversary, ensuring that the requested rental durations of requests are bounded within a finite support.

Assumption 2. *All rental durations are within the range $[d_{\min}, d_{\max}]$, i.e., $d_n \in [d_{\min}, d_{\max}], \forall n \in [N]$.*

In the following, we first prove that obtaining a lossless rounding scheme for a variant of the γ-ocr problem where players have variable rental durations is impossible. Given this, we then develop a new randomized algorithm that uses a new limited-correlation scheme to round the fractional solution, achieving a tight competitive ratio for kRental-Variable.

4.2 Impossibility of 1-ocr under Variable Durations

We show that the correlation scheme of the 1-ocr procedure in Algorithm 2 cannot be extended to settings with variable, request-dependent rental durations; in fact, when the duration of the requested rental intervals varies, it can be proven that no lossless rounding scheme exists.

Theorem 3. *There is no rounding scheme that can attain a lossless rounding, i.e. $\gamma = 1$, for the γ-ocr problem with variable rental durations.*

We prove the theorem by constructing a family of hard instances on which every online algorithm necessarily incurs a non-zero loss. Fundamentally, a lossless algorithm would need to randomize its decisions such that, across all sample paths, the number of allocated items remains balanced. Specifically, upon the arrival of request n, the algorithm must accept it on precisely those sample paths where the number of allocated items is lower than in any other path, or on paths where the number of allocated items is about to decrease due to the imminent expiration of rental durations for previously accepted requests. However, in the variable-duration setting, rental lengths fluctuate unpredictably with the arrival of new requests. A request allocated a unit later than others may return it earlier, introducing significant variability. This makes it infeasible to maintain the necessary correlation across sample paths. As a result, any online algorithm must incur a non-trivial competitive loss. In contrast, when rental durations are fixed, a request that receives a unit later will always return it later. This temporal consistency enables effective correlation of decisions and allows the design of a lossless rounding scheme. In Appendix E, we formally define the γ-ocr problem under variable rental durations and establish the impossibility of a lossless correlation scheme in this setting.

4.3 dop-ϕ-variable: A Randomized Algorithm with Limited Correlation

We present dop-ϕ-variable in Algorithm 4, which also follows a relax-and-round paradigm. Upon the arrival of request n, the algorithm computes the *probabilistic utilization level* $y_n^{(i)}$ for each unit $i \in [k]$. It then selects the unit i_n^* with the lowest utilization level (breaking ties arbitrarily). A fractional allocation $\hat{x}_n \in [0, 1]$ is determined by solving the pseudo-utility maximization problem in Eq. (6), using a pricing function ϕ. Note that here ϕ does not refer to the function in Eq. (4), but instead denotes a new function specifically designed for dop-ϕ-variable. Once $\hat{x}_n$ is computed, the algorithm proceeds to the rounding

phase by correlating the current allocation decision for unit i_n^* with its prior allocation history. Specifically, if unit i_n^* is available, a new random seed is drawn, and the unit is allocated to request n with probability $\hat{x}_n/(1 - y_n^{(i_n^*)})$. This ensures that the overall allocation probability for unit i_n^* is exactly $\hat{x}_n$. *This correlation process is carried out independently for each unit, introducing dependency only among allocation decisions involving the same unit.*

Algorithm 4: Duration-Oblivious Price-based Algorithm with Limited Correlation for kRental-Variable (DOP-ϕ-VARIABLE)

---

1   **Input:** Pricing function $\phi : [0, 1] \rightarrow [d_{\min}, d_{\max}]$.

2   **for** *each arriving request n with (d_n, a_n)* **do**

3   Compute the probabilistic utilization level at a_n: $y_n^{(i)} = \sum_{j=1}^{n-1} \hat{x}_j \cdot \mathbb{I}_{\{a_j+d_j>a_n\}} \cdot \mathbb{I}_{\{i_j^*=i\}}, \forall i \in [k]$.

4   Let $i_n^* = \arg\min_{i \in [k]} \left\{y_n^{(i)}\right\}$ and compute

$$\hat{x}_n = \arg\max_{x \in [0,1]} \ d_n \cdot x - \int_{y_n^{(i_n^*)}}^{y_n^{(i_n^*)}+x} \phi(\eta)d\eta. \tag{6}$$

5   Sample a random seed $S_n \sim U(0, 1)$.

6   **if** $S_n \leq \frac{\hat{x}_n}{1-y_n^{(i_n^*)}}$ *and unit i_n^* available in system* **then**

7          $x_n = 1$ ▷ Allocate unit i_n^* to request n

8   **end**

9   **else**

10         $x_n = 0$.   ▷ Reject request n

11   **end**

12 **end**

Overview of challenges and design principles of ϕ.   Designing a competitive pricing function ϕ for DOP-ϕ-VARIABLE involves two primary challenges. First, unlike the relax-and-round approach adopted for kRental-Fixed, Algorithm 4 integrates the relaxation and rounding steps into a unified procedure, wherein the rounding phase directly influences the structure of the pricing function ϕ. Similar design principles have been employed in related works, such as [12, 8]. Second, as with Algorithm 3 and in contrast to the approaches in [6, 30], Algorithm 4 adopts a myopic, duration-oblivious decision-making strategy. It bases allocation decisions solely on the current utilization levels $\{y_n^{(i)}\}_{i \in [k]}$ at the arrival time a_n of request n, rather than considering the entire rental duration. This duration-oblivious pricing scheme may initially appear counterintuitive, as the pseudo-cost term in Eq. (6) does not depend on the rental duration d_n. Nevertheless, as elaborated in the next section, the pricing function ϕ is carefully designed to implicitly capture potential fluctuations in utilization throughout the entire duration of request n. In essence, our goal is to *maintain a simple and cognitively lightweight pricing scheme, while embedding the complexity into the internal structure of the pricing function ϕ itself.*

4.4   Theoretical Guarantees of DOP-ϕ-VARIABLE

We employ the LP-free certificate method developed by [16] to establish a performance guarantee for DOP-ϕ-VARIABLE. This involves constructing a system of linear constraints parameterized by α, such that the existence of a feasible solution certifies α-competitiveness. Satisfying this system effectively imposes

a distinct constraint for each request n. To address these constraints, we derive a system of differential inequalities that guide the design of the pricing function ϕ. For each possible utilization level at the arrival of request n, we consider the worst-case scenario for utilization fluctuations over its rental interval. Consequently, the design constraints for ϕ naturally arise from this worst-case analysis, embedding such fluctuations explicitly into the pricing function. The following theorem formalizes this result.

Theorem 4. DOP-ϕ-VARIABLE *is α-competitive for the kRental-Variable problem, provided that the pricing function ϕ is increasing and satisfies the following inequalities for all $d_n \in [d_{\min}, d_{\max}]$ and $n \in [N]$:*

$$
\begin{cases}
\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\,\phi(\eta)\,d\eta \;+\; \frac{\alpha}{3}\,d_n\left(\phi^*(d_n) \;-\; 2\,y_1\right) & \geq\; d_n, \quad \forall\, y_1 \in \left[0,\, \frac{\phi^*(d_n)}{2}\right], \\[2mm]
\frac{\alpha}{3}\,d_n\,\phi^*(d_n) \;-\; \frac{\alpha}{3}\,y_2\left(d_n \;-\; \phi(y_2)\right) & \geq\; d_n, \quad \forall\, y_2 \in \left[0,\, \phi^*(d_n)\right],
\end{cases}
\tag{7}
$$

where $\phi^*(d_n) = \sup_{x\in[0,1]}\{\phi(x) \leq d_n\}$.

The detailed proof of the theorem is provided in Appendix F. By Theorem 4, designing an algorithm with guaranteed performance reduces to constructing a pricing function ϕ that satisfies the constraints in Eqs. (7), while minimizing the competitive ratio α. However, solving ϕ is challenging in general because it involves solving a system of *delayed differential inequalities* with an inverse term.

Tightness of results from DOP-ϕ-VARIABLE. Based on Theorem 4, we can show that DOP-ϕ-VARIABLE is $3 \cdot (1 + \ln(\frac{d_{\max}}{d_{\min}}))$-competitive for the kRental-Variable problem when the pricing function ϕ is designed as: $\phi(y) = d_{\min} \cdot \exp([1 + \ln(\frac{d_{\max}}{d_{\min}})] \cdot y - 1), \forall y \in [0, 1]$. This also matches the best-known competitive ratio for kRental-Variable in prior work [15]. On the other hand, we can prove that no online algorithms can attain a competitive ratio smaller than $1 + \ln(d_{\max}/d_{\min})$ for the kRental-Variable problem. Therefore, one simple feasible design of ϕ based on Theorem 4 can already attain an order-optimal competitive ratio for kRental-Variable. We defer all proofs related to the order-optimality to Appendix G.

Comparison to prior work. To demonstrate the significance of our results, we further develop a numerical method in Appendix H to solve Eqs. (7) and obtain a pricing function. In Figure 2, we show that the competitive ratio achieved by DOP-ϕ-VARIABLE (blue curve) surpasses the best known bound of $3(1 + \ln(d_{\max}/d_{\min}))$ from [15].

As a relax and round algorithm, we also investigate the performance of the fractional solution from the relaxation step. Recall that DOP-ϕ-VARIABLE adopts an integrated design of the relaxation and rounding. Thus, deriving the fractional solution without the rounding step needs to additionally modify the LP certificate conditions in Eqs. (7). We show the details in Appendix I and call the resulting new algorithm DOP-ϕ-VARIABLE-FRACTIONAL. The competitive ratio of DOP-ϕ-VARIABLE-FRACTIONAL is illustrated in green curve in Figure 2. We observe that its performance is comparable to $4 + \ln(d_{\max}/d_{\min})$, which is the best-known competitive ratio of kRental-Variable in the large inventory regime (as $k \to \infty$) [6]. We highlight that it is possible to design a new algorithm for the fractional solution of kRental-Variable using the techniques developed in [6] (See Appendix J for more details). In comparison, our proposed DOP-ϕ-VARIABLE-FRACTIONAL uses a far simpler, *duration-oblivious* pseudo-utility maximization rule: the fractional decision for each arriving request depends only on the utilization level at the arrival of the request, whereas the new algorithm based on [6] must additionally track the utilization fluctuation over the entire rental interval of every request.

Furthermore, due to the impossibility results of γ-OCR under variable durations, we are unable to design an online rounding scheme for the fractional solution based on [6]. In contrast, our proposed algorithm DOP-ϕ-VARIABLE integrates the design of deriving the fractional solution and the rounding scheme, which
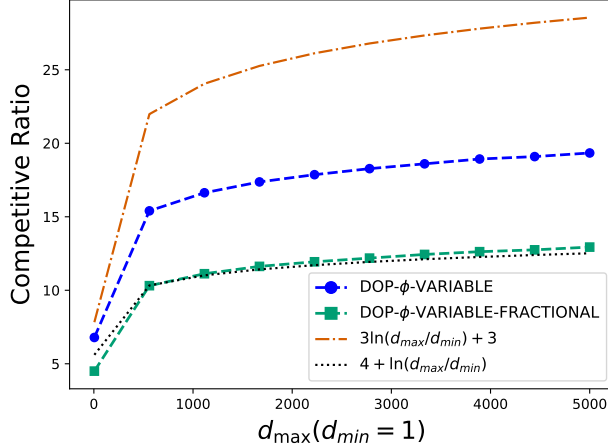
Figure 2: The blue and green curves illustrate the competitive ratios of Algorithm 4 in the integral and fractional settings (using the numerically derived pricing function ϕ). The brown and black curves serve as benchmarks, which correspond to the best known bound for general k [15] and the bound for large k [6].

results in the improved competitive solutions. Designing a tight online rounding scheme for general online fractional solutions remains an interesting open question.

Insights into the hardness of kRental-Variable. To design a pricing function that satisfies the constraints in Theorem 4 with the minimum possible value of α, it suffices to enforce these constraints only at the critical points where the left-hand side of the inequalities is minimized. Furthermore, achieving the optimal value of α requires that the inequalities hold with equality. This condition leads to a system of *delayed differential equations*. Notably, the fact that the pricing function attaining the tightest competitive ratio arises from such a system highlights a *memory* effect: past prices, which are used to reserve a fraction of the resource for short-duration requests, affect the current pricing dynamics. As a result, the pricing function becomes sensitive to the durations of previously accepted requests, particularly those assigned fractional allocations. This sensitivity implies that the pricing function not only sets current prices but also implicitly manages request scheduling by balancing resource reservations in anticipation of returns from short-duration requests. We believe that this provides a new perspective on the fundamental hardness of kRental-Variable and related online allocation problems involving reusable resources, such as in scheduling and matching [18].

5 Conclusion and Future Directions

We studied two adversarial online allocation problems involving reusable resources: the online k-rental problem with fixed rental durations (kRental-Fixed) and a more general variant with variable rental durations (kRental-Variable). To address both settings, we proposed a unified relax-and-round framework that leverages a price-based approach to compute fractional solutions and novel online rounding schemes to convert them into integral decisions. For kRental-Fixed, we integrated the price-based strategy with a new lossless online rounding scheme, 1-ocr, achieving the optimal competitive ratio. In the more general kRental-Variable setting, where lossless rounding is provably unattainable, we developed a limited-correlation rounding strategy that achieves an order-optimal guarantee. We believe that the lossless online

rounding technique introduced in this work is broadly applicable and has the potential to enhance algorithms for a wide range of online resource allocation problems.

This work opens several promising directions for future research. In particular, two open problems stand out as especially compelling: (i) designing more powerful rounding schemes that further narrow the gap between fractional and integral solutions in the variable-duration setting, and (ii) establishing tight upper bounds on the best achievable γ-OCR under variable rental durations, despite the impossibility of attaining 1-OCR as shown by our result.

References

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

[2] Michael O Ball and Maurice Queyranne. Toward robust revenue management: Competitive analysis of online booking. *Operations Research*, 57(4):950–963, 2009.

[3] Ziyun Chen, Zhiyi Huang, Dongchen Li, and Zhihao Gavin Tang. *Prophet Secretary and Matching: the Significance of the Largest Item*, pages 1371–1401.

[4] Steven Delong, Alireza Farhadi, Rad Niazadeh, and Balasubramanian Sivan. Online bipartite matching with reusable resources. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 962–963, New York, NY, USA, 2022. Association for Computing Machinery.

[5] Alon Eden, Michal Feldman, Amos Fiat, and Kineret Segal. An economics-based analysis of ranking for online bipartite matching. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 107–110. SIAM, 2021.

[6] Farbod Ekbatani, Yiding Feng, Ian Kash, and Rad Niazadeh. Online job assignment. 2025.

[7] Ran El-Yaniv, Amos Fiat, Richard M Karp, and Gordon Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30:101–139, 2001.

[8] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. *Journal of the ACM*, 69(6):1–35, 2022.

[9] Ulrich Faigle, Renate Garbe, and Walter Kern. Randomized online algorithms for maximizing busy time interval scheduling. *Computing*, 56(2):95–104, 1996.

[10] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes with applications to bayesian selection problems. *SIAM Journal on Computing*, 50(2):255–300, 2021.

[11] Hu Fu, Pinyan Lu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. Sample-based matroid prophet inequalities. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 781, New York, NY, USA, 2024. Association for Computing Machinery.

[12] Ruiquan Gao, Zhongtian He, Zhiyi Huang, Zipei Nie, Bijun Yuan, and Yan Zhong. Improved online correlated selection. *CoRR*, abs/2106.04224, 2021.

[13] Martin Gardner. Mathematical games. *Scientific american*, 222(6):132–140, 1970.

[14] Minos Garofalakis, Yannis Ioannidis, Banu Özden, and Avi Silberschatz. Competitive on-line scheduling of continuous-media streams. *Journal of Computer and System Sciences*, 64(2):219–248, 2002.

[15] Shashank Goyal and Diwakar Gupta. The online reservation problem. *Algorithms*, 13(10), 2020.

[16] Vineet Goyal, Garud Iyengar, and Rajan Udwani. Asymptotically optimal competitive ratio for online allocation of reusable resources. *Operations Research*, 2024.

[17] Diwakar Gupta and Fei Li. Reserve driver scheduling. *IIE Transactions*, 48(3):193–204, 2016.

[18] Zhiyi Huang, Zhihao Gavin Tang, and David Wajc. Online matching: A brief survey. *SIGecom Exch.*, 22(1):135–158, October 2024.

[19] Zhiyi Huang, Qiankun Zhang, and Yuhao Zhang. Adwords in a panorama. *SIAM Journal on Computing*, 53(3):701–763, 2024.

[20] Tianming Huo and Wang Chi Cheung. Online reusable resource assortment planning with customer-dependent usage durations. SSRN Working Paper, July 9 2023.

[21] Hossein Nekouyan Jazi, Bo Sun, Raouf Boutaba, and Xiaoqi Tan. Posted price mechanisms for online allocation with diseconomies of scale. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*, New York, NY, USA, April 2025. ACM.

[22] Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tightness without counterexamples: A new approach and new results for prophet inequalities. *Mathematics of Operations Research*, 0(0):null, 0.

[23] Zhihao Jiang, Pinyan Lu, Zhihao Gavin Tang, and Yuhao Zhang. Online selection problems against constrained adversary. In *International Conference on Machine Learning*, ICML '21, pages 5002–5012. PMLR, 2021.

[24] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 352–358, New York, NY, USA, 1990. Association for Computing Machinery.

[25] Richard J. Lipton and Andrew Tomkins. Online interval scheduling. In *ACM-SIAM Symposium on Discrete Algorithms*, 1994.

[26] Julian Lorenz, Konstantinos Panagiotou, and Angelika Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

[27] Will Ma. Randomized rounding approaches to online allocation, sequencing, and matching, 2024.

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

[29] Bo Sun, Hossein Nekouyan Jazi, Xiaoqi Tan, and Raouf Boutaba. Static pricing for online selection problem and its variants. In *The 20th Conference on Web and Internet Economics, WINE 2024, Edinburgh, United Kingdom Proceedings ,December 2-5, 2024*. Springer, 2024.

[30] Bo Sun, Lin Yang, Mohammad Hajiesmaili, Adam Wierman, John CS Lui, Don Towsley, and Danny HK Tsang. The online knapsack problem with departures. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS '22)*, 6(3):1–32, 2022.

[31] Xiaoqi Tan, Bo Sun, Alberto Leon-Garcia, Yuan Wu, and Danny HK Tsang. Mechanism design for online resource allocation: A unified approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS '20)*, 4(2):1–46, 2020.

[32] Xiaoqi Tan, Siyuan Yu, Raouf Boutaba, and Alberto Leon-Garcia. Threshold policies with tight guarantees for online selection with convex costs. In *19th International Conference on Web and Internet Economics (WINE '23), Shanghai, China, December 4-8, 2008*. Springer, 2023.

[33] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *4th International Conference on Web and Internet Economics (WINE '08), Shanghai, China, December 17-20, 2008. Proceedings 4*, pages 566–576. Springer, 2008.

Appendix

A Proof of Proposition 1

Let $P_n = \{i \in [n-1] \mid a_i + d > a_n\}$ denote the set of players whose rental intervals overlap with that of player n. Let E_n denote the event that a ball is allocated to player n. Define A_i as the event that the random seed $s_n \leq f_k \cdot \hat{x}_n$ for the n-th player, for each $i \in [N]$. For convenience in notation, we occasionally abuse the notation by using E_n or A_n to also denote the indicator random variable $\mathbb{I}_{E_n}$ and $\mathbb{I}_{A_n}$. Then, we have:

$$\mathbb{P}[E_n] = \mathbb{P}[A_n] \cdot \mathbb{P}\left[\sum_{i \in P_n} E_i < k\right] \geq \mathbb{P}[A_n] \cdot \mathbb{P}\left[\sum_{i \in P_n} A_i < k\right] = f_k \cdot \hat{x}_n \cdot \mathbb{P}\left[\sum_{i \in P_n} A_i < k\right].$$

The inequality $\mathbb{P}\left[\sum_{i \in P_n} E_i < k\right] \geq \mathbb{P}\left[\sum_{i \in P_n} A_i < k\right]$ holds because, whenever $\sum_{i \in P_n} A_i < k$ occurs, the event $\sum_{i \in P_n} E_i < k$ must also occur. This is true since if fewer than k players satisfy the seed condition (which means that they are eligible for allocation), then at most $k-1$ players can have a resource unit under allocation at the arrival of player n. Thus, to prove the proposition, it suffices to lower bound the term $\mathbb{P}\left[\sum_{i \in P_n} A_i < k\right]$ and show that this probability is at least $1 - \exp\left(-\frac{(k - f_k \cdot k)^2}{f_k \cdot k + k}\right)$.

The random variables $\{A_i\}_{i \in P_n}$ are mutually independent since the random seed s_i for each player i is sampled independently from the uniform distribution.

Furthermore, we have

$$\mathbb{E}\left[\sum_{i \in P_n} A_i\right] = \sum_{i \in P_n} \mathbb{E}[A_i] = \sum_{i \in P_n} f_k \cdot \hat{x}_i \leq f_k \cdot k,$$

where the last inequality follows from the condition in Eq. (1).

We can restate an alternative form of the Chernoff bound as follows:

Lemma 2 (Chernoff Bound). *Let $\{X_i\}_{i=1}^{n}$ be independent random variables taking values in $[0,1]$, and let $S = \sum_{i=1}^{n} X_i$ with mean $\mu = \mathbb{E}[S]$. Then, for any threshold $A > \mu$, $\mathbb{P}[S \geq A] \leq \exp\left(-\frac{(A-\mu)^2}{A+\mu}\right)$.*

Applying the above form of Chernoff bound gives

$$\mathbb{P}\left[\sum_{i \in P_n} A_i \geq k\right] \leq \exp\left(-\frac{(k - \mathbb{E}[\sum_{i \in P_n} A_i])^2}{\mathbb{E}[\sum_{i \in P_n} A_i] + k} \cdot\right) \leq \exp\left(-\frac{(k - f_k \cdot k)^2}{f_k \cdot k + k}\right),$$

where the second inequality follows from the inequality $\mathbb{E}[\sum_{i \in P_n} A_i] \leq f_k \cdot k$ and the fact that function $\exp\left(-\frac{(k-x)^2}{x+k}\right)$ is increasing when $x \leq k$ and $f_k < 1$. Putting together everything, we have

$$\mathbb{P}[E_n] = \mathbb{P}[A_n] \cdot \mathbb{P}[\sum_{i \in P_n} A_i < k] \leq \hat{x}_n \cdot f_k \cdot \left(1 - \exp\left(-\frac{(k - f_k \cdot k)^2}{f_k \cdot k + k}\right)\right).$$

We thus complete the proof of Proposition 1.

B Proofs and Examples for 1-OCR

B.1 Proof of Proposition 2

For each player $n \in [N]$ and ball $i \in [k]$, we define the set $R_n^{(i)}$ as follows:

$$R_n^{(i)} = \left\{ \left(p_j, \max\{1, p_j + \hat{x}_j\}, j\right) \big| j \in B_n^{(i)} \right\} \cup \left\{ \left(0, p_j + \hat{x}_j - 1, j\right) \big| j \in B_n^{(i-1)}, p_j + \hat{x}_j > 1 \right\}, \tag{8}$$

where the set of players in $B_n^{(i)}$ are defined as

$$B_n^{(i)} = \left\{ j \in [n] \mid a_j + d > a_n, m_j = i, \hat{x}_j \neq 0 \right\}.$$

For each ball i, the set $R_n^{(i)}$ contains all triplets that each correspond to the process of the algorithm deciding to allocate ball i to player n. Specifically, if an element of the form $\left(p_j, \max\{1, p_j + \hat{x}_j\}, j\right)$ appears in $R_n^{(i)}$ and $j \in B_n^{(i)}$, then upon the arrival of player j, if the random seed r lies in the range $(p_j, \max\{1, p_j + \hat{x}_j\}]$, Algorithm 2 allocates ball $m_j = i$ to player j.

Similarly, if $\left(0, p_j + \hat{x}_j - 1, j\right)$ belongs to $R_n^{(i)}$ and $j \in B_n^{(i-1)}$, then upon the arrival of player j, if $r \in [0, p_j + \hat{x}_j - 1)$, Algorithm 2 allocates ball $m_j + 1 = i$ to player j. Thus, the element $\left(0, p_j + \hat{x}_j - 1, j\right)$ is added to $R_n^{(i)}$ to reflect this allocation decision.

In what follows, we show that if Proposition 2 fails, then there must exist two elements (s_1, e_1, n_1) and (s_2, e_2, n_2) in either $R_n^{(m_n)}$ or $R_n^{(m_{n+1})}$ whose intervals $[s_1, e_1)$ and $[s_2, e_2)$ intersect. We consider the following two cases, reflecting those discussed in Proposition 2:

- **Case 1:** $p_n + \hat{x}_n < 1$. If Proposition 2 holds, then whenever $r \in [p_n, p_n + \hat{x}_n)$, ball m_n remains available at the arrival of player n. Suppose the proposition fails in this case. Then, for some $r \in [p_n, p_n + \hat{x}_n)$, ball m_n is unavailable. Hence, there must exist a player $j < n$ such that if $r \in [s_1, e_1)$, with $[s_1, e_1) \cap [p_n, p_n + \hat{x}_n) \neq \emptyset$, ball m_n was allocated to j. Thus, in this case Proposition 2 does not hold and by the definitions of $R_n^{(m_n)}$, the set $R_n^{(m_n)}$ contains two triplets (s_1, e_1, j) and $(p_n, p_n + \hat{x}_n, n)$ such that range $[s_1, e_1)$ and $[p_n, p_n + \hat{x}_n)$ overlap.

- **Case 2:** $p_n + \hat{x}_n \geq 1$. A similar argument shows that if Proposition 2 fails in this case, there must exist two elements (s_1, e_1, n_1) and (s_2, e_2, n_2) in either $R_n^{(m_n)}$ or $R_n^{(m_{n+1})}$ where the intervals $[s_1, e_1)$ and $[s_2, e_2)$ intersect.

Next, we show that no set $R_n^{(i)}$ (for any $n \in [N]$ and $i \in [k]$) can contain two elements whose corresponding intervals overlap. Indeed, Proposition 4 establishes that $R_n^{(i)}$ can only be in one of four specific forms, none of which allows two elements (s_1, e_1, n_1) and (s_2, e_2, n_2) with overlapping intervals $[s_1, e_1)$ and $[s_2, e_2)$. Let us define $l_n^{(i)} = \sum_{(s,e,t) \in R_n^{(i)}} (e - s)$.

Proposition 4. *At the arrival time of each player $n \in [N]$ and for each ball $i \in [k]$, if the set $R_n^{(i)}$ is non-empty, then upon sorting $R_n^{(i)}$ by the first element of each triplet, one of the following four forms arises:*
(i) If $i = m_n$ and $\hat{x}_n + p_n < 1$, then for some $j \in \mathbb{N}$, $R_n^{(i)}$ takes the form

$$R_n^{(i)} = \left\{ (s_1, s_2, t_1), \ldots, (s_j = p_n, s_{j+1} = p_n + \hat{x}_n, t_j), (s_{j+1} + 1 - l_n^{(i)}, s_{j+2}, t_{j+1}), \ldots, (s_q, s_{q+1} = 1, t_q) \right\},$$

where $t_{j+1} < t_{j+2} < \cdots < t_q < t_1 < t_2 < \cdots < t_j$ and $s_1 \leq s_2 \leq \cdots \leq s_{j+1} \leq s_{j+2} \leq \cdots \leq s_{q+1} = 1$.

(ii) *If $i = m_n$ and $\hat{x}_n + p_n \geq 1$, $R_n^{(i)}$ takes the form*

$$R_n^{(i)} = \left\{ (s_1, s_2, t_1), (s_2, s_3, t_2) \ldots, (s_q = p_n, s_{q+1} = 1, t_q) \right\},$$

where $t_1 < t_2 < \cdots < t_q$ and $s_1 \leq s_2 \leq \cdots s_q = p_n \leq s_{q+1} = 1$.

(iii) *If $i = m_n + 1$ and $\hat{x}_n + p_n > 1$, then $R_n^{(i)}$ takes the form*

$$R_n^{(i)} = \left\{ (s_1 = 0, s_2 = p_n + \hat{x}_n - 1, t_1 = n), (s_2, s_3, t_2), (s_3, s_4, t_3), \ldots, (s_q, s_{q+1} = 1, t_q) \right\},$$

where $t_2 < \cdots < t_q < t_1 = n$ and $s_1 = 0 < s_2 < \cdots < s_{q+1}$.

(iv) *If $i = m_n + 1$ and $\hat{x}_n + p_n \leq 1$, or if $i \in [k] - \{m_n, m_n + 1\}$, then $R_n^{(i)}$ takes the form*

$$R_n^{(i)} = \left\{ (s_1 = 1 - l_n^{(i)}, s_2, t_1), (s_2, s_3, t_2), \ldots, (s_q, s_{q+1} = 1, t_q) \right\},$$

where $t_1 < t_2 < \cdots < t_q$ and $1 - l_n^{(i)} = s_1 \leq s_2 \leq \cdots \leq s_{q+1} = 1$.

Proof. We prove the proposition by induction on N, the number of arriving players.

Base case ($N = 1$). In this scenario, we have $p_1 = 0$ and $m_1 = 1$, leading to $R_1^{(1)} = \{(0, x_1, 1)\}$. If $\hat{x}_n + p_n < 1$ then $R_1^{(1)}$ takes the first form $j = 1$, otherwise it is in the second form and all the other sets $R_1^{(i)}$ are empty. Therefore, the proposition holds for $N = 1$.

Induction step: Assume that for $N = M$ the proposition holds, i.e., for each $i \in [k]$, the set $R_M^{(i)}$ is in one of the above forms in Proposition 4. We now prove that $R_{M+1}^{(i)}$ is also in one of these forms after processing the $(M + 1)$-th player. After the arrival of the $(M + 1)$-th player, two types of changes may occur in $R_{M+1}^{(i)}$ compared to $R_M^{(i)}$: existing elements may be removed, or new elements may be added. An element in $R_M^{(i)}$ is removed from $R_{M+1}^{(i)}$ if the rental period for the corresponding player ends by time a_{M+1}. In other words, for an element corresponding to a player j satisfying $a_M < a_j + d \leq a_{M+1}$, that element is removed from $R_{M+1}^{(i)}$. In the following discussion, we examine the two scenarios that can arise depending on whether an element is added or removed from the set $R_M^{(i)}$.

B.1.1 Scenario 1 (element removed)

Suppose an element is removed from $R_M^{(i)}$ upon the arrival of player $M + 1$. We consider four subcases, depending on which form $R_M^{(i)}$ takes.

• **Subcase-1a: $R_M^{(i)}$ is in the first form.** Then we have

$$R_M^{(i)} = \{(s_1 = 0, s_2, t_1), \ldots, (s_j = p_n, s_{j+1} = p_n + \hat{x}_n, t_j), (s_{j+1} + 1 - l_n^{(i)}, s_{j+2}, t_{j+1}), \ldots, (s_q, s_{q+1} = 1, t_q)\}.$$

for some values $\{s_i\}$ and $\{t_i\}$ satisfying the conditions of the first form in Proposition 4.

In the set $R_M^{(i)}$, the first elements to be removed—based on the arrival times of these players and the fact that they appear in $R_M^{(i)}$ only for a fixed duration—are those corresponding to players $j + 1$ up to q and 1 to j. It is easy to see that the removal of the elements corresponding to players $j + 1$ up to q leads $R_{M+1}^{(i)}$ to remain in the first form. The removal of elements corresponding to players 1 to j causes $R_{M+1}^{(i)}$ to have either the first or second form, depending on the values of x_{m+1} and p_{m+1}.

• **Subcase-1b:** $R_M^{(i)}$ **is in the second form.** Then we have

$$R_M^{(i)} = \{(s_1, s_2, t_1), (s_2, s_3, t_2), \ldots, (s_q = p_M, s_{q+1} = p_{M+1}, t_q)\},$$

for some values $\{s_i\}$ and $\{t_i\}$ satisfying the conditions of the second form in Proposition 4.

Since $t_1 \leq t_i$ for all $i \in [q]$, (s_1, s_2, t_1) is the oldest element and thus the first to be removed. Consequently, if one element is removed upon the arrival of player $M+1$, it must be (s_1, s_2, t_1). A straightforward check shows that after this removal, $R_{M+1}^{(i)}$ continues to be in the fourth form. If more than one element is removed, a similar argument implies that $R_{M+1}^{(i)}$ will be in the fourth form.

• **Subcase-1c:** $R_M^{(i)}$ **is in the third form.** If an element is removed from $R_M^{(i)}$ upon the arrival of player $M + 1$, it must be the oldest element in that set. By an argument analogous to the subcases above, $R_{M+1}^{(i)}$ then transitions to either the first or second form depending on the values of x_{m+1} and p_{m+1}.

• **Subcase-1d:** $R_M^{(i)}$ **is in the fourth form.** Here, removing an element from $R_M^{(i)}$ at the arrival of player $M + 1$ causes $R_{M+1}^{(i)}$ to remain in the same form.

B.1.2 Scenario 2 (element is added)

Suppose a new element corresponding to the decision made for player $M + 1$ is added to the set $R_{M+1}^{(i)}$. Again, we consider the following cases:

• **Subcase-2a:** $R_M^{(i)}$ **in the first form.** In this case, we have

$$R_M^{(i)} = \{(s_1 = 0, s_2, t_1), \ldots, (s_j = p_M, s_{j+1} = p_{M+1}, t_j), (s_{j+1} + 1 - l_M^{(i)}, s_{j+2}, t_{j+1}), \ldots, (s_q, s_{q+1} = 1, t_q)\}.$$

Suppose a new element $(s, e, M + 1)$ is added at the arrival time of the $(M + 1)$-th player. By the design of Algorithm 2, we have $s = p_{M+1}$ and $e = p_{M+1} + x_{M+1}$. Thus, this new element is inserted after the element $(s_j, s_{j+1} = p_{M+1}, t_j)$. To ensure that $R_{M+1}^{(i)}$ remains in the proper form (i.e., either in the first or second form), we must have

$$e \leq s_{j+1} + 1 - l_M^{(i)}.$$

We now formalize and prove this requirement.

Lemma 3. *Consider the case where the set $R_M^{(i)}$ is in the first form as described above, and suppose that upon the arrival of player $M + 1$, a new element $(s, e, M + 1)$ is added with $s = p_{M+1}$ and $e = p_{M+1} + x_{M+1}$. Then, it holds that*

$$e \leq s_{j+1} + 1 - l_M^{(i)},$$

where $l_M^{(i)} = \sum_{(s,e,t) \in R_M^{(i)}} (e - s)$, as previously defined, represents the total sum of the range sizes corresponding to each element (s, e, t) in $R_M^{(i)}$.

Proof. Consider the stage at which the algorithm has completed processing the t_q-th player and has added the element $(s_q, s_{q+1} = 1, t_q)$ to $R_{t_q}^{(i)}$. By construction, at this point the pointer p_{t_q+1} reaches 1, and the pointer m_{t_q} is advanced. Between the processing of the t_q-th and t_1-th players, the pointer m_n is advanced $k - 1$ times until it once again points to ball i, at which point a new element (s_1, s_2, t_1) is added to $R_{t_1}^{(i)}$. For

the pointer m_n to be advanced $k-1$ times, there must exist a sequence of players from the t_q-th to the t_1-th such that the following condition holds:

$$\sum_{j=t_q+1}^{t_1-1} \hat{x}_j + \left(x_{t_q} + p_{t_q} - 1\right) + \left(1 - p_{t_1}\right) \cdot \mathbf{1}_{\{m_{t_1}=i-1\}} = k - 1.$$

Next, it follows that if

$$e > s_{j+1} + 1 - l_M^{(i)},$$

then

$$\sum_{j=1}^{M+1} \hat{x}_j \cdot \mathbf{1}_{\{a_j+d>a_{M+1}\}} > k,$$

which contradicts the constraint in Eq. (1). Therefore, we must have

$$e \le s_{j+1} + 1 - l_M^{(i)}.$$

Thus, we complete the proof of Lemma 3. $\qquad\qquad\square$

Thus, by Lemma 3, if a new element is added to $R_{M+1}^{(i)}$ when $R_M^{(i)}$ is in the first form, no two elements will have overlapping intervals, and the set remains in either the first or second form. This completes the argument.

• **Subcase-2b: $R_M^{(i)}$ in the second form.** Since at the arrival time of player M we have $i = m_n$, if a new element is added to $R_{M+1}^{(i)}$ at the arrival time of player $M + 1$, it must be that $k = 1$. In this case, the newly added element takes the form $(p_{M+1}, p_{M+1} + x_{M+1}, M + 1)$. By the same reasoning, if the interval $(p_{M+1}, p_{M+1}+x_{M+1}]$ overlaps with any other element in $R_{M+1}^{(i)}$, this would contradict the constraint specified in Eq. (1).

• **Subcase-2c: $R_M^{(i)}$ in the third form.** In this case, we have

$$R_M^{(i)} = \left\{ \left(s_1 = 0, s_2 = p_M + x_M - 1, t_1 = M\right), \left(s_2, s_3, t_2\right), \dots, \left(s_q, s_{q+1} = 1, t_q\right) \right\}.$$

Suppose a new element (s, e, t) is added at the arrival time of player $M+1$. Then, by the design of Algorithm 2,

$$s = p_M + x_M - 1 = p_{M+1},$$

$$e = \max\{1, p_{M+1} + x_{M+1}\}.$$

Thus, this new element is appended after the element (s_1, s_2, M). Following the same reasoning as in Subcase-1a, we must have $e < s_2$, otherwise a contradiction arises with the constraint in Eq. (1).

• **Subcase-2d: $R_M^{(i)}$ in the fourth form.** Here, we have

$$R_M^{(i)} = \left\{ \left(s_1 = 1 - l_M^{(i)}, s_2, t_1\right), \left(s_2, s_3, t_2\right), \dots, \left(s_q, s_{q+1} = 1, t_q\right) \right\}.$$

Suppose a new element (s, e, t) is added at time $M + 1$. By the design of Algorithm 2, we have

$$s = 0 = p_{M+1},$$

$$e = \max\{1, p_{M+1} + x_{M+1}\}.$$

Hence, this new element is inserted before (s_1, s_2, t_1). Again, by analogous reasoning to Subcase-1a, we must have $e < s_2$; otherwise, we reach a contradiction with the constraint in Eq. (1).

Putting everything together, we complete the proof of Proposition 4. □

Our above proof of Proposition 4 shows that $R_n^{(i)}$ can only take one of four specific forms, none of which permits two elements (s_1, e_1, n_1) and (s_2, e_2, n_2) with overlapping intervals $[s_1, e_1)$ and $[s_2, e_2)$. Thus, Proposition 2 follows.

B.2 Proof of Theorem 1

We consider two cases based on the relation between $1 - p_n$ and $\hat{x}_n$.

Case I: $1 - p_n \geq \hat{x}_n$. In this case, the algorithm assigns ball m_n to player n if ball m_n is available and $r \in [p_n, p_n + \hat{x}_n)$. Let E_{m_n} denote the event that ball m_n is available upon the arrival of the n-th player at time a_n, and let E denote the event $r \in [p_n, p_n + \hat{x}_n)$. Then the probability that ball m_n is allocated to player n is

$$\Pr[E_{m_n} \cap E].$$

Based on the definition of the set $R_n^{(m_n)}$, it contains all the ranges that have been used to allocate ball m_n to previous players, including those whose rental players overlap with player n. Proposition 4 ensures that the range $[p_n, p_n + \hat{x}_n)$ does not overlap with any other range in $R_n^{(m_n)}$. Thus, if the random sample $r \in [p_n, p_n + \hat{x}_n)$, then ball m_n is guaranteed to be available at the arrival of player n. So if E occurs, ball m_n is available. Consequently,

$$\begin{aligned}
&\Pr[E_{m_n} \cap E] \\
&= \Pr[E] \cdot \Pr[E_{m_n} \mid E] \\
&= \Pr[E] \\
&= \Pr[r \in [p_n, p_n + \hat{x}_n)] \\
&= \hat{x}_n.
\end{aligned}$$

Hence, when $\hat{x}_n < 1 - p_n$, player n gets ball m_n with probability $\hat{x}_n$.

Case II: $1 - p_n < \hat{x}_n$. In this case, the algorithm first attempts to allocate ball m_n if it is available and if $r \in [p_n, 1]$. If this allocation does not occur, it then attempts to allocate ball $m_n + 1$ provided that it is available and $r \in [0, \hat{x}_n - 1 + p_n)$. Let E_{m_n} denote the event that ball m_n is available at time a_n, and let E_1 denote the event $r \in [p_n, 1]$. Similarly, let E_{m_n+1} denote the event that ball $m_n + 1$ is available, and let E_2 denote the event $r \in [0, \hat{x}_n - 1 + p_n)$. Then the total probability that a ball is allocated to player n is given by

$$\Pr[E_{m_n} \cap E_1] + \Pr\left[E_{m_n+1} \cap E_2 \cap (E_{m_n} \cap E_1)'\right].$$

Based on Proposition 4 and the fact that $p_n \neq 0$, the set $R_n^{(m_n)}$ is in the first form and the range $[p_n, 1]$ does not overlap with any previous ranges in $R_n^{(m_n)}$. Therefore, if E_1 occurs, ball m_n is available, and

$$\Pr[E_{m_n} \cap E_1] = \Pr[E_1] = 1 - p_n.$$

Similarly, the event $E_{m_n+1} \cap E_2$ occurs with probability

$$\Pr[E_{m_n+1} \cap E_2] = \Pr[E_2] = \hat{x}_n - 1 + p_n.$$

Since the ranges $[p_n, 1]$ and $[0, \hat{x}_n - 1 + p_n)$ do not overlap (because $\hat{x}_n \leq 1$), we have

$$
\begin{aligned}
&\Pr\Big[E_{m_n+1} \cap E_2 \cap (E_{m_n} \cap E_1)'\Big] \\
&= \Pr\Big[r \in \big([0, \hat{x}_n - 1 + p_n) \cap [0, p_n)\big)\Big] \\
&= \Pr\Big[r \in [0, \hat{x}_n - 1 + p_n)\Big] \\
&= \hat{x}_n - 1 + p_n.
\end{aligned}
$$

Thus, the total probability that a ball is allocated to player n is

$$
\begin{aligned}
&\Pr[\text{ball allocated to player } n] \\
&= \Pr[E_{m_n} \cap E_1] + \Pr\Big[E_{m_n+1} \cap E_2 \cap (E_{m_n} \cap E_1)'\Big] \\
&= (1 - p_n) + (\hat{x}_n - 1 + p_n) \\
&= \hat{x}_n.
\end{aligned}
$$

Thus, in both cases, each player n gets a ball with the desired probability $\hat{x}_n$, which proves that Algorithm 2 is a lossless online scheme.

B.3 Example: How 1-OCR Works in Algorithm 2

To illustrate how Algorithm 2 works and demonstrate how it outperforms a rounding procedure that makes independent decisions at each time step, consider the following instance with $k = 2$, $d = 5$, and $N = 4$. The target probability values and arrival time of players are given by:

$$\Big\{(a_1 = 1, \hat{x}_1 = 0.4), \quad (a_2 = 2, \hat{x}_2 = 0.5), \quad (a_3 = 3, \hat{x}_3 = 0.6), \quad (a_4 = 6, \hat{x}_4 = 0.6)\Big\}.$$

It can be verified that for each player $n \in \{1, \cdots, 4\}$, the inequality in Eq. (1) is satisfied for the above instance. Initially, before any player arrives, a random seed r is drawn from the uniform distribution $U(0, 1)$. This random seed is used throughout the horizon to correlate the algorithm's decisions.

- When player 1 arrives at time $t = 1$ with $\hat{x}_1 = 0.4$, Algorithm 2 assigns the first ball to player 1 if $r \in [0, 0.4)$. Thus, **player 1 will receive one ball with probability 0.4**.

- When player 2 arrives at time $t = 2$ with $\hat{x}_2 = 0.5$, Algorithm 2 assigns the first ball provided that $r \in [0.4, 0.9)$. Thus, player 2 receives the first ball with probability 0.5. Note that if $r \in [0.4, 0.9)$, the first ball is guaranteed to be available because it was rented at $t = 1$ only if $r \in [0, 0.4)$. Thus, **player 2 will receive one ball with probability 0.5**.

- When player 3 arrives at time $t = 3$ with $\hat{x}_3 = 0.6$, Algorithm 2 assigns the first ball if it is available and $r \in [0.9, 1]$, which happens with probability 0.1. If the first ball is not allocated, Algorithm 2 assigns the second ball if $r \in [0, 0.5)$. Consequently, **the total probability of allocating one ball to player 3 is 0.6**.
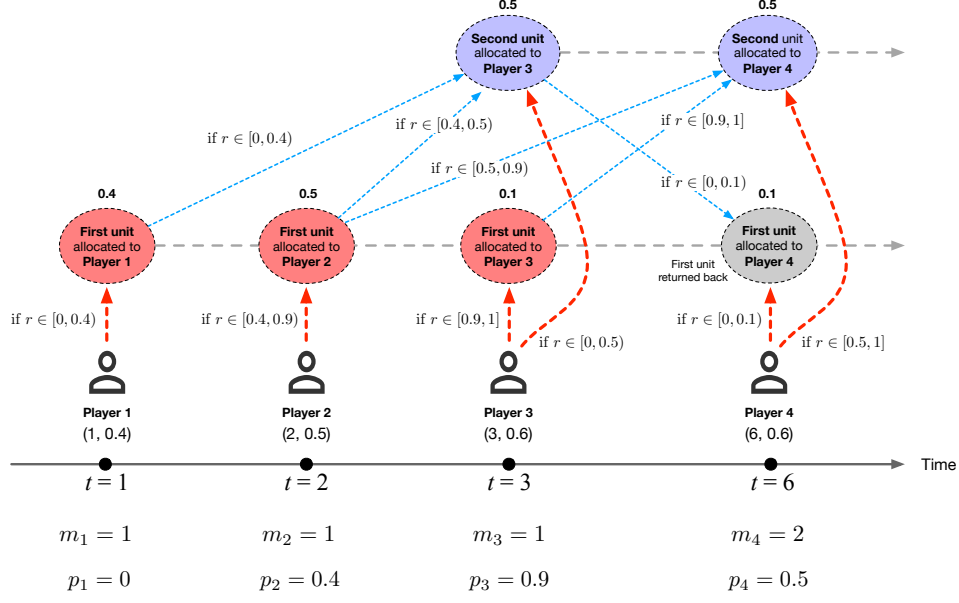
Figure 3: Illustration of the online correlated assignment process in Algorithm 2. The number above each circle represents the probability of the corresponding event occurring. In this example, player 3 receives the first ball with probability 0.1 and the second ball with probability 0.5. Notably, the assignment of the second ball to player 3 (with probability 0.5) is correlated with the decisions made for players 1 and 2, as illustrated by the two blue dashed arrows on the left. For instance, if the realization of the random seed is $r = 0.45$, then player 1 is rejected, player 2 receives the first ball, player 3 receives the second ball, and player 4 is rejected.

- Finally, when player 4 arrives at time $t = 6$ with $\hat{x}_4 = 0.6$, Algorithm 2 assigns the second ball if $r \in [0.5, 1]$, an event that occurs with probability 0.5; assigns the first ball if $r \in [0, 0.1)$, an event that occurs with probability 0.1. Note that for the case when $r \in [0, 0.1)$, the first ball will be returned by player 1 at $t = 6$. Thus, **player 4 will receive one ball with probability 0.6**.

We can see from the above example that Algorithm 2 always maintains a 1- OCR scheme in that it assigns one ball to every player $n \in [N]$ with probability exactly $\hat{x}_n$.

C Proof of Theorem 2

We use an online primal-dual approach to establish the competitive ratio of Algorithm 3. Consider the dual LP corresponding to the primal LP in Eq. (2):

$$\min_{\lambda, \mathbf{u}} \quad \sum_{n \in [N]} u_n + k \cdot \sum_{n \in [N]} \lambda_n, \tag{9}$$

$$\text{s.t.} \quad u_n + \sum_{j=n}^{N} \lambda_j \mathbf{1}_{\{a_n + d > a_j\}} \geq v_n, \quad \forall n \in [N], \tag{10}$$

$$\lambda_n \geq 0, \quad \forall n \in [N]. \tag{11}$$

Let $\mathsf{ALG}(\mathcal{I})$ represent the expected objective value of Algorithm 3 on an instance $\mathcal{I}$. Moving forward, we first propose a feasible solution for the primal problem given in Eq. (2), denoted as $\{x_n^{\mathsf{ALG}}\}_{n \in [N]}$. We

then show that $\mathsf{ALG}(\mathcal{I})$ equals the objective value of the primal LP solution.

In the second step, we design a feasible solution for the dual problem, denoted as $\{u_n^{\mathsf{ALG}}, \lambda_n^{\mathsf{ALG}}\}_{n \in [N]}$, which corresponds to the dual linear program, and define the dual objective

$$D^{\mathsf{ALG}} = \sum_{n \in [N]} u_n^{\mathsf{ALG}} + k \cdot \sum_{n \in [N]} \lambda_n^{\mathsf{ALG}}.$$

We then establish that the dual solution $\{u_n^{\mathsf{ALG}}, \lambda_n^{\mathsf{ALG}}\}_{n \in [N]}$ is feasible for the dual LP in Eq. (9).

In the final step, we establish that

$$D^{\mathsf{ALG}} \leq \underbrace{\left(1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)\right)}_{F} \cdot P^{\mathsf{ALG}}.$$

After proving these steps, by weak duality we have

$$\mathsf{ALG}(\mathcal{I}) = P^{\mathsf{ALG}} \geq \frac{1}{F} \cdot D^{\mathsf{ALG}} \geq \frac{1}{F} \cdot \mathsf{OPT}(\mathcal{I}),$$

where $\mathsf{OPT}(\mathcal{I})$ denotes the objective value of the optimal clairvoyant algorithm on instance $\mathcal{I}$. Thus, the $1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$-competitiveness of Algorithm 3 follows.

Step-I: Design of primal solution and $\mathsf{ALG}(\mathcal{I}) = P^{\mathsf{ALG}}$. We set the primal variables as $x_n^{\mathsf{ALG}} = \hat{x}_n$ for all $n \in [N]$. Let $\Delta_{P^{\mathsf{ALG}}}^{(n)}$ denote the increase in the primal LP objective value resulting from updating x_n^{ALG}, and let $\Delta_n^{\mathsf{ALG}(\mathcal{I})}$ denote the corresponding increase in the expected objective value of Algorithm 3 when a unit of resource is allocated to request n. By our update, we have

$$\Delta_{P^{\mathsf{ALG}}}^{(n)} = v_n \cdot x_n^{\mathsf{ALG}}.$$

Furthermore,

$$\Delta_n^{\mathsf{ALG}(\mathcal{I})} = v_n \cdot \Pr[\text{a unit of resource is allocated to request } n]$$
$$= v_n \cdot \hat{x}_n.$$

The final equality follows because the 1-OCR rounding scheme allocates a unit to request n with probability exactly $\hat{x}_n$. Summing over requests for all $n \in [N]$ establishes that $\mathsf{ALG}(\mathcal{I}) = P^{\mathsf{ALG}}$.

Step-II: Design of dual solution, $\{u_n^{\mathsf{ALG}}, \lambda_n^{\mathsf{ALG}}\}_{n \in [N]}$, and feasibility of the dual solution. Let us initialize all dual variables to zero. For each rental request n, let $\hat{x}_n$ be the fractional allocation chosen by Algorithm 3. We then perform the following updates:

$$\lambda_{v_n^*}^{\mathsf{ALG}} = \lambda_{v_n^*}^{\mathsf{ALG}} + F \cdot \int_{\eta = \frac{y_n}{k}}^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta, \tag{12}$$

$$u_n^{\mathsf{ALG}} = F \cdot \hat{x}_n \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right), \tag{13}$$

where $v_n^* = \max\{j \geq n | a_j < a_n + d\}$ and $F = 1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$. To prove the feasibility of the above dual solution, we must show that the dual constraint in Eq. (10) corresponding to each request n is satisfied; that is,

$$u_n^{\mathsf{ALG}} + \sum_{j=n}^{N} \lambda_j^{\mathsf{ALG}} \cdot \mathbb{I}_{\{a_n + d > a_j\}} \geq v_n.$$

For the n-th request, the dual variable u_n^{ALG} is updated according to Eq. (13), so that

$$u_n^{\text{ALG}} \geq F \cdot \hat{x}_n \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right).$$

Next, we aim to prove that

$$\sum_{j=n}^{N} \lambda_j^{\text{ALG}} \cdot \mathbb{I}_{\{a_n + d > a_j\}} \geq F \cdot \int_0^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta. \tag{14}$$

Assuming this inequality holds, we first show that the dual constraint in Eq. (10) is satisfied for each request n and then get back to the proof of the above inequality. Let us consider the following two cases.

Case 1: $\phi\left(\frac{y_n + \hat{x}_n}{k}\right) < v_n$. It can be verified that following from Eq. (3), we have

$$\hat{x}_n = \max\left\{0, \min\left\{1, k \cdot \phi^{-1}(v_n) - y_n\right\}\right\}.$$

Since we have $\phi\left(\frac{y_n + \hat{x}_n}{k}\right) < v_n$, then it follows that $\hat{x}_n = 1$. Therefore,

$$u_n^{\text{ALG}} + \sum_{j=n}^{N} \lambda_j^{\text{ALG}} \cdot \mathbb{I}_{\{a_n + d > a_j\}} \geq F \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right) + F \cdot \int_0^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta$$

$$\geq F \cdot \left(\int_{\frac{y_n + \hat{x}_n}{k}}^{\phi^{-1}(v_n)} \phi(\eta) d\eta + \int_0^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta\right)$$

$$= F \cdot \int_0^{\phi^{-1}(v_n)} \phi(\eta) d\eta$$

$$\geq v_n,$$

where the second inequality follows from the monotonicity of ϕ, and the last inequality follows from the design of ϕ in Theorem 2.

Case 2: $\phi\left(\frac{y_n + \hat{x}_n}{k}\right) \geq v_n$. We can lower bound the LHS of Eq. (10) as follows:

$$u_n^{\text{ALG}} + \sum_{j=n}^{N} \lambda_j^{\text{ALG}} \cdot \mathbb{I}_{\{a_n + d > a_j\}} \geq F \cdot x_n \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right) + F \cdot \int_0^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta$$

$$\geq F \cdot \int_0^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta) d\eta$$

$$\geq F \cdot \int_0^{\phi^{-1}(v_n)} \phi(\eta) d\eta$$

$$\geq v_n,$$

where the second inequality follows from the monotonicity of ϕ and the last inequality is as in Case 1.

Thus, assuming Eq. (14) holds, the dual constraint in Eq. (10) is satisfied for each $n \in [N]$, proving the feasibility of the dual solution $\{u_n^{\text{ALG}}, \lambda_n^{\text{ALG}}\}$.

Let us now return to the proof of inequality Eq. (14). Define the set of requests O_n as

$$O_n = \{n \leq j \leq N \mid a_n + d > a_j, \hat{x}_j > 0\},$$

27

i.e., the set of requests arriving after request n whose rental request intervals overlap with that of request n. Furthermore, define the set of requests B_n as

$$B_n = \{1 \le j \le n \mid a_j + d > a_n, \hat{x}_j > 0\}.$$

Note that for each request $j \in B_n$, we have $v_j^* \in O_n$. Then, it follows that:

$$\sum_{j=n}^{N} \lambda_j^{\text{ALG}} \cdot \mathbb{I}_{\{a_n+d>a_j\}} \ge \sum_{j \in O_n} \lambda_j^{\text{ALG}}$$

$$= \sum_{j \in O_n} \sum_{m \in [n]} \Delta_m^{\lambda_j^{\text{ALG}}}$$

$$\ge \sum_{m \in B_n} \Delta_m^{\lambda_{v_m^*}^{\text{ALG}}},$$

where $\Delta_m^{\lambda_j^{\text{ALG}}}$ denotes the increase in the value of λ_j^{ALG} due to the dual update in Eq. (13) performed for request m. In the above inequalities, the first inequality follows from the definition of the set O_n and the second inequality follows from the fact that the request v_m^* is in the set O_n for each request $m \in B_n$.

Moving forward, let us sort the requests in B_n in increasing order of their arrival times so that

$$B_n = \{n_1, n_2, \ldots, n_{|B_n|}\}.$$

We have:

$$\sum_{m \in B_n} \Delta_m^{\lambda_{v_m^*}^{\text{ALG}}} \ge F \sum_{i=1}^{|B_n|} \int_{\eta=y_{n_i}}^{y_{n_i}+\hat{x}_{n_i}} \phi(\eta) d\eta$$

$$\ge F \sum_{i=1}^{|B_n|} \int_{\eta=\sum_{j=1}^{i-1} x_{n_j}}^{\sum_{j=1}^{i} \hat{x}_{n_j}} \phi(\eta) d\eta$$

$$= F \int_{\eta=0}^{\sum_{i=1}^{|B_n|} \hat{x}_{n_i}} \phi(\eta) d\eta$$

$$= F \int_{\eta=0}^{y_n+\hat{x}_n} \phi(\eta) d\eta.$$

Here, the first inequality follows from the update for $\lambda_{v_m^*}^{\text{ALG}}$ in Eq. (13) for each request m, and the second inequality holds because, by the definition of the set B_n, all requests in B_n arriving prior to request n_i have rental requests overlapping request n_i; hence, the variable y_{n_i} is at least $\sum_{j=1}^{i-1} x_{n_j}$ at the arrival of request n_i.

Putting together the results obtained in the above two cases, the inequality in Eq. (14) follows.

Step-III: Proof of $D^{\text{ALG}} \le F \cdot P^{\text{ALG}}$. Instead of directly proving the overall inequality $D^{\text{ALG}} \le F \cdot P^{\text{ALG}}$, we show that for each request n:

$$\Delta_n^{D^{\text{ALG}}} \le F \Delta_n^{P^{\text{ALG}}},$$

where $\Delta_n^{D^{\text{ALG}}}$ denotes the increase in the dual objective value after updating the dual variables for request n via Eqs. (12) and (13), and $\Delta_n^{P^{\text{ALG}}}$ denotes the increase in the primal objective value after updating the primal LP solution for request n by setting $x_n^{\text{ALG}} = \hat{x}_n$.

We now consider the following two cases in order to prove the above inequality.

- **Case 1: $\hat{x}_n = 1$.** In this case, we have

$$\Delta_n^{D^{\text{ALG}}} = \Delta_n^{u_n^{\text{ALG}}} + k\Delta_n^{\lambda_{v_n^*}^{\text{ALG}}}$$

$$= F \cdot \hat{x}_n \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right) + kF \cdot \int_{\eta = \frac{y_n}{k}}^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta)d\eta$$

$$= F \cdot \left(v_n - \phi\left(\frac{y_n + 1}{k}\right)\right) + kF \cdot \int_{\eta = \frac{y_n}{k}}^{\frac{y_n + 1}{k}} \phi(\eta)d\eta$$

$$\leq F \cdot \left(v_n - \phi\left(\frac{y_n + 1}{k}\right)\right) + kF \cdot \frac{1}{k}\phi\left(\frac{y_n + 1}{k}\right)$$

$$= F \cdot v_n = F\Delta_n^{P^{\text{ALG}}},$$

where the first equality follows from the objective value of the dual LP in Eq. (9). The second equality follows from the dual updates done in Eq. (12) and Eq. (13). In addition, the first inequality follows from the fact that ϕ is an increasing function.

- **Case 2: $\hat{x}_n < 1$.** In this case, if $\hat{x}_n = 0$, then we have $\Delta_n^{D^{\text{ALG}}} = \Delta_n^{P^{\text{ALG}}} = 0$. Otherwise, if $\hat{x}_n \neq 0$, then by Eq. (3) we have

$$\hat{x}_n = \max\left\{0, \min\left\{1, k \cdot \phi^{-1}(v_n) - y_n\right\}\right\}.$$

Since $\hat{x}_n < 1$, we must have

$$v_n = \phi\left(\frac{y_n + \hat{x}_n}{k}\right).$$

It follows that

$$\Delta_n^{D^{\text{ALG}}} = \Delta_n^{u_n^{\text{ALG}}} + k\Delta_n^{\lambda_{v_n^*}^{\text{ALG}}}$$

$$= F \cdot \hat{x}_n \cdot \left(v_n - \phi\left(\frac{y_n + \hat{x}_n}{k}\right)\right) + kF \cdot \int_{\eta = \frac{y_n}{k}}^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta)d\eta$$

$$= kF \cdot \int_{\eta = \frac{y_n}{k}}^{\frac{y_n + \hat{x}_n}{k}} \phi(\eta)d\eta,$$

$$\leq Fk \cdot \frac{\hat{x}_n}{k} \cdot \phi\left(\frac{y_n + \hat{x}_n}{k}\right)$$

$$= F \cdot v_n \cdot \hat{x}_n$$

$$= F\Delta_n^{P^{\text{ALG}}}.$$

Here, the inequality follows from the fact that the function ϕ is increasing. Combining the results from both cases, we obtain

$$\Delta_n^{D^{\text{ALG}}} \leq F\Delta_n^{P^{\text{ALG}}},$$

which completes the proof of the third step and thus the proof for Theorem 2.

D Proof of Proposition 3

Following the same proof structure as in [29] for the online selection problem, we can prove the lower bound $1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$ on the competitiveness of every online algorithm for kRental-Fixed.

We design a set of hard instances for the kRental-Fixed problem similar to [29]. Let $\mathcal{A}(k, v)$ denote a batch of k identical requests, each with valuation v (with $v \in [v_{\min}, v_{\max}]$). Divide the uncertainty range $[v_{\min}, v_{\max}]$ into $m - 1$ sub-ranges of equal length

$$\Delta_v = \frac{v_{\max} - v_{\min}}{m - 1}.$$

Let

$$\mathcal{V} := \{v_i\}_{i \in [m]},$$

where $v_i = v_{\min} + (i - 1)\Delta_v$ for $i \in [m]$. Define an instance

$$I_{v_i} := \mathcal{A}(k, v_1) \oplus \mathcal{A}(k, v_2) \oplus \cdots \oplus \mathcal{A}(k, v_i),$$

which consists of a sequence of request batches with increasing valuations that arrive consecutively within an arbitrarily short time interval. (Here, $\mathcal{A}(k, v_i) \oplus \mathcal{A}(k, v_j)$ denotes a batch $\mathcal{A}(k, v_i)$ immediately followed by a batch $\mathcal{A}(k, v_j)$.) In this construction, all requests, including those from the first batch $\mathcal{A}(k, v_1)$ to the last request in the final batch $\mathcal{A}(k, v_m)$, arrive during the short interval $[0, \epsilon]$, where ϵ is a small value satisfying $\epsilon < d$. This setup guarantees that if a unit is allocated to a request in batch $\mathcal{A}(k, v_i)$, it cannot be reallocated to any request arriving afterwards, because the unit will only become available after d time steps, while subsequent requests arrive within a much shorter time span.

We consider the collection $\{I_{v_i}\}_{i \in [m]}$ as the set of hard instances for the kRental-Fixed problem. Following the same proof structure as in the proof of Lemma 2.3 in [29], to obtain the optimal online algorithm on the set of hard instances $\{I_{v_i}\}_{i \in [m]}$, one can show that the lower bound $1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$ holds for the competitiveness of every online algorithm on the set of hard instances $\{I_{v_i}\}_{i \in [m]}$ and thus prove the lower bound $1 + \ln\left(\frac{v_{\max}}{v_{\min}}\right)$.

E Impossibility Result for the Existence of a Lossless Rounding in the Variable Duration Setting

Consider the γ-OCR problem introduced in Section 2.1, where each player rents a ball for a fixed duration d that is identical across all players and independent of their identity. In this setting, the rental period is uniform and known in advance.

In the following, we study a variant of this problem in which the rental duration is player-dependent and may vary across players. Specifically, each player requests a rental duration that is revealed only upon their arrival. Thus, the algorithm receives information about a player's rental duration in an online fashion. A more formal definition of the OCR problem under this variable-duration setting is as follows.

Definition 2 (γ-OCR-V). *Consider a set of k identical balls, each uniquely labeled from the set $\{1, 2, \ldots, k\}$. Each ball can be assigned to a player for a variable duration, determined by the player's requested rental period, after which the ball becomes available for reassignment. A sequence of N players arrives one by one, with each player n characterized by a triple $(a_n, \hat{x}_n, d_n)$, where a_n is the arrival time, $\hat{x}_n \in [0, 1]$ is the target probability with which a ball should be assigned to player n, and d_n is the requested rental duration. For any $\gamma \in [0, 1]$, a γ-OCR-V algorithm, given the input sequence $\{(\hat{x}_n, a_n, d_n)\}_{n \in [N]}$, assigns a ball to each player*

$n \in [N]$ with probability at least $\gamma \hat{x}_n$. Similar to the definition of γ-OCR, for any given input instance of γ-OCR-V, $\{(\hat{x}_n, a_n, d_n)\}_{n \in [N]}$, we impose the following condition:

$$\hat{x}_n \leq \min\left(1, k - \sum_{j \in [n-1]} \hat{x}_j \cdot \mathbb{I}_{\{a_j + d_j > a_n\}}\right), \qquad \forall n \in [N]. \tag{15}$$

Proof of Theorem 3 We proceed by contradiction. Suppose, for the sake of contradiction, that a (possibly randomized) online algorithm ALG is *lossless* for the γ-OCR-V problem; that is, it achieves $\gamma = 1$ on every instance. Consider the following instance with inventory of $k = 2$ balls and:

$$\mathcal{I} = \left\{ \underbrace{(a_1 = 1, x_1 = 0.5, d_1 = 5)}_{P_1}, \underbrace{(a_2 = 2, x_2 = 0.5, d_2 = 7)}_{P_2}, \underbrace{(a_3 = 5.5, x_3 = \tfrac{2}{3}, d_3 = 9)}_{P_3}, \right.$$

$$\left. \underbrace{(a_4 = 6, x_4 = \tfrac{1}{3}, d_4 = 8)}_{P_4}, \underbrace{(a_5 = 8, x_5 = \tfrac{1}{2}, d_5 = 10)}_{P_5}, \underbrace{(a_6 = 14, x_6 = \tfrac{5}{6}, d_6 = 10)}_{P_6} \right\}.$$

The above instance clearly satisfies the constraint in Eq. (15). Let $A_n^{(i)}$ ($i = 1, 2$) be the event that the i-th ball is allocated to player n by the online algorithm ALG upon processing the player's request. Likewise, let $E_t^{(i)}$ ($i = 1, 2$) denote the event that the i-th ball is still available at time t.

Next, we prove the following claim:

Lemma 4. *For the algorithm ALG to be lossless, we must have:*

$$\mathbb{P}[A_1^{(1)} \cap A_2^{(2)}] + \mathbb{P}[A_1^{(2)} \cap A_2^{(1)}] = 0$$

Proof. We prove the claim by contradiction. Suppose the lemma fails, i.e.,

$$\mathbb{P}[A_1^{(1)} \cap A_2^{(2)}] + \mathbb{P}[A_1^{(2)} \cap A_2^{(1)}] > 0.$$

Then the probability that *at least one* of the two balls is still available immediately after the decision for player 2 satisfies

$$\mathbb{P}[E_2^{(1)} \cup E_2^{(2)}] = 1 - \mathbb{P}[(A_1^{(1)} \cup A_2^{(1)}) \cap (A_1^{(2)} \cup A_2^{(2)})]$$
$$= 1 - \mathbb{P}[(A_1^{(1)} \cap A_1^{(2)}) \cup (A_2^{(1)} \cap A_2^{(2)})]$$
$$= 1 - \mathbb{P}[A_1^{(1)} \cap A_2^{(2)}] - \mathbb{P}[A_1^{(2)} \cap A_2^{(1)}] < 1,$$

where the strict inequality follows from the contradictory assumption.

Next, let us create a new instance $\mathcal{I}'$ that is identical to $\mathcal{I}$ for players P_1 and P_2, but replaces the remaining sequence by a single player P_3' with

$$(a_3 = 3, x_3 = 1, d_3 = 10).$$

The feasibility condition (15) is easily verified for $\mathcal{I}'$.

Because the first two players in $\mathcal{I}$ and $\mathcal{I}'$ are identical, algorithm ALG follows the same random path on both instances for the first two players. Thus, at $a_3 = 3$ in $\mathcal{I}'$ we still have $\mathbb{P}[E_2^{(1)} \cup E_2^{(2)}] < 1$, meaning that with probability less than one either of balls are available at arrival of request P_3'. Yet a lossless algorithm must allocate a ball to P_3' with probability $x_3 = 1$, an impossibility. Hence our initial assumption is false, and the lemma follows. □

Next, consider the arrival of the third player P_3 in instance $\mathcal{I}$, with parameters $a_3 = 5$, $x_3 = \frac{2}{3}$, and $d_3 = 9$. We prove the following statement.

Lemma 5. *For* ALG *to remain lossless on* $\mathcal{I}$*, it must hold that*

$$\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] = \frac{1}{6}.$$

Proof. First, let us prove that

$$\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] \leq \frac{1}{6}. \tag{16}$$

The proof is again by contradiction. Assume that the above inequality does not hold.

Consider the instance $\mathcal{I}'$, in which the first three players are identical to those of instance $\mathcal{I}$, but the fourth player P_4' is characterised as $(a_4 = 7, d_4 = 10, x_4 = \frac{5}{6})$. It can be verified that the feasibility constraint in Eq. (1) is satisfied for every player. Because the first three arrivals in $\mathcal{I}$ and $\mathcal{I}'$ are identical, the online algorithm ALG behaves the same on both instances up to arrival of P_4'. Consequently, when P_4' arrives in $\mathcal{I}'$,

$$\mathbb{P}[E_7^{(1)} \cup E_7^{(2)}] = 1 - \mathbb{P}[(A_2^{(1)} \cup A_3^{(1)}) \cap (A_2^{(2)} \cup A_3^{(2)})]$$

$$= 1 - \mathbb{P}[(A_2^{(1)} \cap A_3^{(2)}) \cup (A_3^{(1)} \cap A_2^{(2)})]$$

$$= 1 - \mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] - \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] < \frac{5}{6},$$

where the last strict inequality is by the contradictory assumption to have $\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] > \frac{1}{6}$. The allocation of P_1 does not influence the calculation above, because by $a_4 = 7$ any ball assigned to P_1 has already returned to the system. Hence, at the arrival of player P_4' in $\mathcal{I}'$ the probability that either ball is still available, $\mathbb{P}[E_7^{(1)} \cup E_7^{(2)}]$, is strictly less than $\frac{5}{6}$. Therefore, ALG cannot allocate a ball to P_4 with the required target probability $\frac{5}{6}$, contradicting the lossless property. Thus we must have

$$\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] \leq \frac{1}{6}.$$

Next, let us prove that

$$\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] \geq \frac{1}{6},$$

after which we can conclude that the sum equals $\frac{1}{6}$.

First, we compute the following probability:

$$\mathbb{P}[(A_1^{(1)} \cup A_1^{(2)}) \cup (A_2^{(1)} \cup A_2^{(2)})] = \mathbb{P}[(A_1^{(1)} \cup A_1^{(2)})] + \mathbb{P}[(A_2^{(1)} \cup A_2^{(2)})]$$

$$- \mathbb{P}[(A_1^{(1)} \cup A_1^{(2)}) \cap (A_2^{(1)} \cup A_2^{(2)})]$$

$$= x_1 + x_2 - \mathbb{P}[(A_1^{(1)} \cap A_2^{(2)}) \cup (A_1^{(2)} \cap A_2^{(1)})]$$

$$= 1 - \mathbb{P}[(A_1^{(1)} \cap A_2^{(2)})] - \mathbb{P}[(A_1^{(2)} \cap A_2^{(1)})]$$

$$= 1,$$

where the last equality follows from Lemma 4. By the same reasoning we have

$$\mathbb{P}[(A_1^{(1)} \cup A_1^{(2)}) \cap (A_2^{(1)} \cup A_2^{(2)})] = 0.$$

32

Since $\mathbb{P}[A_3^{(1)} \cup A_3^{(2)}] = x_3 = \frac{2}{3}$, $\mathbb{P}[(A_1^{(1)} \cup A_1^{(2)})] = \frac{1}{2}$, and $\mathbb{P}[(A_1^{(1)} \cup A_1^{(2)}) \cup (A_2^{(1)} \cup A_2^{(2)})] = 1$, we must have

$$\mathbb{P}[(A_3^{(1)} \cup A_3^{(2)}) \cap (A_2^{(1)} \cup A_2^{(2)})] \geq \frac{1}{6}.$$

Combining this with the earlier inequality establishes

$$\mathbb{P}[A_2^{(1)} \cap A_3^{(2)}] + \mathbb{P}[A_2^{(2)} \cap A_3^{(1)}] = \frac{1}{6}.$$

Thus, the proof is completed. □

Next, consider the arrival of player P_4 in instance $\mathcal{I}$, with parameters $a_4 = 6$, $x_4 = \frac{1}{3}$, and $d_4 = 7$.

Lemma 6. *For* ALG *to be lossless, after the arrival of player P_4 we must have*

$$\mathbb{P}[A_3^{(1)} \cap A_4^{(2)}] + \mathbb{P}[A_3^{(2)} \cap A_4^{(1)}] = 0.$$

Proof. In case the above equality is not satisfied, a request P_5 could arrive at time $a_5 = 10$ with target probability $x_5 = 1$, and ALG would not be lossless since $\mathbb{P}[E_{10}^{(1)} \cup E_{10}^{(2)}] < 1$. □

Next, consider the arrival of player P_5 at time $a_5 = 8$, with target probability $x_5 = \frac{1}{2}$ and $d_5 = 10$. Let us refer to the time right before the arrival of fifth player as 8^-.

Lemma 7. *After the arrival of player P_5,*

$$\mathbb{P}[A_3^{(1)} \cap A_5^{(2)}] + \mathbb{P}[A_3^{(2)} \cap A_5^{(1)}] \geq \frac{1}{2}.$$

Proof. From the previous two lemmas 5 and 6, we have

$$\mathbb{P}[(A_2^{(1)} \cup A_2^{(2)}) \cap (A_3^{(1)} \cup A_3^{(2)})] = \frac{1}{6}, \qquad \mathbb{P}[(A_3^{(1)} \cup A_3^{(2)}) \cap (A_4^{(2)} \cup A_4^{(1)})] = 0.$$

Let us define C_1, C_2, and C_3 as follows:

$$C_1 = \left(A_3^{(1)} \cup A_3^{(2)}\right) \cap \left(A_2^{(2)} \cup A_2^{(1)}\right)',$$

$$C_2 = \left(A_2^{(2)} \cup A_2^{(1)}\right) \cap \left(A_3^{(1)} \cup A_3^{(2)}\right)',$$

$$C_3 = \left(A_3^{(1)} \cup A_3^{(2)}\right) \cap \left(A_2^{(2)} \cup A_2^{(1)}\right).$$

These events are pairwise disjoint and $\mathbb{P}[C_1 \cup C_2 \cup C_3] = 1$.

Claim 1. $\mathbb{P}[E_{8^-}^{(1)} \cup E_{8^-}^{(2)} \mid C_2] = 0$.

Proof. From Lemma 6 we have

$$\mathbb{P}[(A_3^{(1)} \cup A_3^{(2)}) \cap (A_4^{(2)} \cup A_4^{(1)})] = 0,$$

thus it follows that $\mathbb{P}[C_1 \cap (A_4^{(2)} \cup A_4^{(1)})] = 0$ and $\mathbb{P}[C_3 \cap (A_4^{(2)} \cup A_4^{(1)})] = 0$. Note that we have $\mathbb{P}[A_4^{(2)} \cup A_4^{(1)}] = \mathbb{P}[C_2] = \frac{1}{3}$, so $A_4^{(2)} \cup A_4^{(1)} = C_2$. Next,

$$
\begin{aligned}
C_2 &= C_2 \cap C_2 \\
&= \left((A_4^{(2)} \cup A_4^{(1)}) \cap \left((A_2^{(2)} \cup A_2^{(1)}) \cap (A_3^{(1)} \cup A_3^{(2)})' \right) \right) \\
&= \left((A_4^{(2)} \cup A_4^{(1)}) \cap (A_2^{(2)} \cup A_2^{(1)}) \right) \cap \left((A_4^{(2)} \cup A_4^{(1)}) \cap (A_3^{(1)} \cup A_3^{(2)})' \right) \\
&= \left((A_4^{(2)} \cup A_4^{(1)}) \cap (A_2^{(2)} \cup A_2^{(1)}) \right) \cap (A_4^{(2)} \cup A_4^{(1)}) \\
&= \left((A_4^{(2)} \cup A_4^{(1)}) \cap (A_2^{(2)} \cup A_2^{(1)}) \right) \\
&= (A_2^{(2)} \cap A_4^{(1)}) \cup (A_2^{(1)} \cap A_4^{(2)}),
\end{aligned}
$$

where the third equality uses $\mathbb{P}[(A_3^{(1)} \cup A_3^{(2)}) \cap (A_4^{(2)} \cup A_4^{(1)})] = 0$. Based on the above equations, under the event C_2, either $A_2^{(2)} \cap A_4^{(1)}$ or $A_2^{(1)} \cap A_4^{(2)}$ occurs, so neither ball 1 nor ball 2 is available. Therefore

$$
\mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_2] = 0.
$$

This completes the proof. $\qquad\square$

Next, we can verify that
$$
\mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_3] = 0.
$$
In fact, $C_3 = (A_3^{(1)} \cup A_3^{(2)}) \cap (A_4^{(2)} \cup A_4^{(1)}) = (A_3^{(1)} \cap A_4^{(2)}) \cup (A_3^{(2)} \cap A_4^{(1)})$. Thus, under C_3, either $(A_3^{(1)} \cap A_4^{(2)})$ or $(A_3^{(2)} \cap A_4^{(1)})$ occurs, so neither ball 1 nor ball 2 is available, and $\mathbb{P}[E_4^{(1)} \cup E_4^{(2)} \mid C_3] = 0$.

Since at the arrival of P_5 we have $\sum_{i \in [4]} x_i \cdot \mathbb{I}_{\{a_i + d_i \geq a_5\}} = \frac{3}{2}$, with probability $\frac{1}{2}$ one of the two balls is available, and

$$
\begin{aligned}
\frac{1}{2} &= \mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)}] \\
&= \mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_1] + \mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_2] + \mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_3] \\
&= \mathbb{P}[E_{8-}^{(1)} \cup E_{8-}^{(2)} \mid C_1] = \mathbb{P}[C_1].
\end{aligned}
$$

Thus, we have $E_{8-}^{(1)} \cup E_{8-}^{(2)} = C_1$.

At the arrival of P_5, if either ball is available, the other ball is already allocated to P_3, since $C_1 = (A_3^{(1)} \cup A_3^{(2)}) \cap (A_2^{(2)} \cup A_2^{(1)})'$. Therefore, a lossless $\mathtt{ALG}$ must allocate one of the two balls to P_5 with probability $\frac{1}{2}$, and

$$
\begin{aligned}
&\mathbb{P}[(A_5^{(1)} \cup A_5^{(2)}) \cap (A_3^{(1)} \cup A_3^{(2)})] \\
&= \mathbb{P}[(A_3^{(1)} \cap A_5^{(2)}) \cup (A_3^{(2)} \cup A_5^{(1)})] \\
&= \mathbb{P}[A_3^{(1)} \cap A_5^{(2)}] + \mathbb{P}[A_3^{(2)} \cup A_5^{(1)}] \\
&= \frac{1}{2}.
\end{aligned}
$$

This completes the proof. $\qquad\square$

Now consider the arrival of the sixth player with $(a_6 = 13.5, x_6 = \frac{5}{6}, d_6 = 10)$. It can be verified that

$$\mathbb{P}[E_6^{(1)} \cup E_6^{(2)}] = 1 - \mathbb{P}[(A_3^{(1)} \cup A_5^{(1)}) \cap (A_3^{(2)} \cup A_5^{(2)})]$$

$$= 1 - \mathbb{P}[(A_3^{(1)} \cap A_5^{(2)}) \cup (A_3^{(2)} \cap A_5^{(1)})]$$

$$= \frac{1}{2},$$

where the last equality follows from Lemma 7. Thus, at the arrival of player P_6, with probability equal to $\frac{1}{2}$ either ball is available, and the algorithm cannot allocate a ball with target probability $\frac{5}{6}$ losslessly to player P_6. We thus complete the proof.

F Proof of Theorem 4

To establish the α-competitiveness of Algorithm 4, we apply the LP-free certificate approach developed [16]. We define a system of linear constraints that acts as a certificate of α-competitiveness when feasible. The variables in this system include $\{\{u_n, \lambda_n^{(i)}\}_{n \in [N]}, \theta_i\}_{i \in [k]}$. Conceptually, we employ a pricing scheme inspired by the economic interpretation of the randomized primal-dual framework in [5]. In this setting, u_n denotes the utility of request n, while $\lambda_n^{(i)}$ represents the price of unit i at the time of request n's arrival. The variable θ_i corresponds to the revenue obtained by the algorithm from selling the i-th unit of the resource. The constraints are given by:

$$\sum_{n \in [N]} u_n + \sum_{i \in [k]} \theta_i \leq \alpha \cdot \mathsf{ALG}(I), \tag{17}$$

$$\theta_i + \sum_{n \in P_i} u_n \geq \sum_{n \in P_i} d_n = \mathsf{OPT}_i, \qquad \forall i \in [k], \tag{18}$$

where $\mathsf{ALG}(I)$ denotes the expected performance of the algorithm on instance I. For each i, the set P_i consists of the requests to whom the optimal clairvoyant algorithm allocates the i-th unit of the resource, and $\mathsf{OPT}_i = \sum_{n \in P_i} d_n$. Clearly, if the above constraints hold, then:

$$\mathsf{OPT} = \sum_{i=1}^{k} \mathsf{OPT}_i$$

$$\leq \sum_{i=1}^{k} \left(\theta_i + \sum_{n \in P_i} u_n \right)$$

$$\leq \sum_{i \in [k]} \theta_i + \sum_{n \in [N]} u_n$$

$$\leq \alpha \cdot \mathsf{ALG}(I),$$

where in the derivation above, the first inequality follows from Eq. (17), the second inequality holds because the sets $\{P_i\}$ are pairwise disjoint, and the final inequality follows from Eq. (18). Hence, if there is a solution to this linear system, it certifies the α-competitiveness of Algorithm 4. In what follows, we first describe how to assign values to the system's variables, and then show that these assignments satisfy all the inequalities of the system.

Based on the performance of Algorithm 4, we now specify how to assign values to the variables in our LP-free certificate. Initially, all variables are set to zero. After the arrival of the final request in instance I,

we update the variables as follows. For each rental request n, let $\hat{x}_n$ be the fractional allocation variable determined by Algorithm 5. We then perform the following updates:

$$u_n = \frac{\alpha}{3} d_n \hat{x}_n, \tag{19}$$

$$\lambda_j^{(i_n^*)} = \lambda_j^{(i_n^*)} + \frac{\alpha}{3}\left(a_{j+1} - a_j\right)\hat{x}_n, \quad \forall j, n \le j < v_n^*, a_j < a_n + d_n, \tag{20}$$

$$\lambda_{v_n^*}^{(i_n^*)} = \lambda_{v_n^*}^{(i_n^*)} + \frac{\alpha}{3}\left(2d_n - \left(a_{v_n^*} - a_n\right)\right)\hat{x}_n, \tag{21}$$

where $v_n^* = \max\{j \ge n \mid a_j < a_n + d_n\}$. After updating the variables above for all the N requests, let us set the value of variables $\theta_i = \sum_{n\in[N]} \lambda_n^{(i_n^*)}$ for all $i \in [k]$.

Remark 1. *The intuition behind the design of the variables in the system is as follows. We implement a pricing scheme in which the variable u_n denotes the utility accrued by request n, and the set of variables $\lambda_j^{(i)}$ represents the price of unit i at the arrival time of request j. Depending on the fractional allocation $\hat{x}_n$, which is allocated from unit i_n^* to request n, the utility u_n increases accordingly. Furthermore, we raise the price of unit i_n^* at the arrival time of every request $j < v_n^*$ whose request interval overlaps with that of request n. This price adjustment is intended for any request arriving after request n whose rental duration finishes before that of request n and who might be rejected because a fraction of unit i_n^* has already been allocated to request n. In addition, we also increase the price of unit i_n^* for request v_n^*, which is the last arriving request in the horizon whose request interval overlaps with that of request n. This adjustment accounts for requests who might arrive very close to the end of request n's request interval and whose requests may be rejected because the algorithm has already allocated a fraction of unit i_n^* to request n.*

First constraint of the system. Based on the construction of the variables detailed above, we now verify the constraint Eq. (17).

Let Δ_n^{ALG} denote the increase in the expected objective value of Algorithm 4 after processing request n. We show that $\Delta_n^{\text{ALG}} = d_n \cdot \hat{x}_n$. Algorithm 4 first computes $y_n^{(i)}$, the probabilistic utilization level of item i, which sums the probabilities that unit i has been allocated to previous requests. Consequently, at the arrival of request n, resource i is available with probability $1 - y_n^{(i)}$. Therefore,

$$\begin{aligned}
\Delta_n^{\text{ALG}} &= d_n \cdot \mathbb{P}[\text{unit } i_n^* \text{ is available}] \cdot \mathbb{P}[\text{the unit is allocated}] \\
&= d_n\left(1 - y_n^{(i_n^*)}\right) \cdot \frac{\hat{x}_n}{1 - y_n^{(i_n^*)}} \\
&= d_n \hat{x}_n.
\end{aligned}$$

Summing over all requests $n \in [N]$, $\text{ALG}(I) = \sum_{n\in[N]} \Delta_n^{\text{ALG}} = \sum_{n\in[N]} d_n \hat{x}_n$.

Next, let Δ_n^{RHS} denote the increase in the right-hand side of Eq. (17) after updating the variables of the system according to Eqs. (19)–(21) for the n-th rental request. It follows directly that

$$\Delta_n^{\text{RHS}} = \alpha d_n \hat{x}_n.$$

Summing these increments for all requests $n \in [N]$ gives

$$\sum_{n\in[N]} u_n + \sum_{i\in[k]} \theta_i = \sum_{n\in[N]} \Delta_n^{\text{RHS}} = \sum_{n\in[N]} \alpha \cdot d_n \cdot \hat{x}_n.$$

Combining this with our previous result on the expected increase in the objective of the algorithm, we conclude that the first constraint, Eq. (17), is satisfied.

Second set of constraints. The remaining set of constraints are as follows:

$$\theta_i + \sum_{n \in P_i} u_n \geq \alpha \cdot \sum_{n \in P_i} d_n = \alpha \cdot \mathsf{OPT}_i,$$

where the value of θ_i can be lower bounded by:

$$\theta_i = \sum_{n \in [N]} \lambda_n^{(i)} \geq \sum_{n \in P_i} \sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}}.$$

The inequality holds because, for each request $n \in P_i$, the set $\{j \in [N] \mid a_n + d_n > a_j\}$ cannot intersect with any other request's set. Otherwise, the optimal offline algorithm would have allocated unit i to two requests at the same time, which is infeasible. Hence, for the left-hand side of Eq. (18), we have

$$\theta_i + \sum_{n \in P_i} u_n \geq \sum_{n \in P_i} \sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}} + \sum_{n \in P_i} u_n \geq \sum_{n \in P_i} \left(\sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}} + u_n \right).$$

Thus, it suffices to show that for each request $n \in P_i$,

$$\sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}} + u_n \geq \alpha \cdot d_n.$$

In what follows, we first derive a lower bound on the left-hand side of the above inequality by separately bounding $\sum_{j=n}^{N} \lambda_j^{(i)} \cdot \mathbb{I}_{\{a_n + d_n > a_j\}}$ and u_n. Then, invoking the constraints placed on the price function ϕ in Theorem 4, we conclude that this bound is at least $\alpha \cdot d_n$, thus ensuring that Eq. (18) is satisfied.

Let us define

$$\Lambda_n^{(i)} = \sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}}.$$

Let us further define the set of requests $B_n^{(i)}$ by

$$B_n^{(i)} = \left\{ 1 \leq j < n \mid a_j + d_j > a_n, \hat{x}_j > 0, i_j^* = i \right\}. \tag{22}$$

In other words, $B_n^{(i)}$ is the set of all requests who arrive before request n, request rental intervals overlapping the interval of request n, and have a non-zero probability of receiving unit i.

Furthermore, define $C_n^{(i)} \subseteq B_n^{(i)}$ by

$$C_n^{(i)} = \left\{ j \in B_n^{(i)} \mid a_j + d_j < a_n + d_n \right\}.$$

Next, let the set of requests $\{c_l\}_{l \in [L]}$ be defined recursively such that

$$c_1 = \underset{j \in C_n^{(i)}}{\arg \min} \{a_j + d_j\},$$

$$c_l = \underset{\substack{j \in C_n^{(i)} \\ a_j > a_{c_{l-1}}}}{\arg \min} \{a_j + d_j\}, \qquad 1 < l \leq L.$$

The request $c_L \in C_n^{(i)}$ is the one for which $\{j \in C_n^{(i)} \mid a_j > a_{c_L}\}$ is empty. Using the sequence of requests $\{c_l\}_{l \in [L]}$, we partition $C_n^{(i)}$ into L sets:

$$C_l = \{j \in C_n^{(i)} \mid a_{c_{l-1}} \le a_j < a_{c_l}\}, \quad \forall l \in [L].$$

Furthermore, define the set of values $\{z_l\}_{l \in [L]}$ by

$$z_l = \sum_{\substack{j \in C_n^{(i)} \\ a_j < a_{c_l}}} \hat{x}_j.$$

Let $\Delta_{\Lambda_n^{(i)}}^{(j)}$ denote the increase in $\Lambda_n^{(i)}$ after processing the rental request of request j and updating the variables according to Eqs. (19)–(21). From the way we perform the updates, it follows that

$$\sum_{j \in C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)} \ge \sum_{l=1}^{L} \sum_{j \in C_l} \frac{\alpha}{3} \hat{x}_j \Big(d_j + (a_j + d_j - a_n) \Big)$$

$$\ge \sum_{l=1}^{L} \sum_{j \in C_l} \frac{\alpha}{3} \hat{x}_j \Big(d_{c_l} + (a_{c_l} + d_{c_l} - a_n) \Big)$$

$$\ge \sum_{l=1}^{L} \frac{\alpha}{3} (z_l - z_{l-1}) \Big(2 d_{c_l} + a_{c_l} - a_{c_1} \Big),$$

where the first inequality follows from the dual updates in Eqs. (20)–(21), the second inequality relies on the definition of c_l and C_l (under which $d_j \le d_{c_l}$ and $a_j + d_j \ge a_{c_l} + d_{c_l}$ for each $j \in C_l$), and the last inequality follows from the definition of the values $\{z_l\}_{l \in [L]}$, taking $z_0 = 0$.

Define the sequence $\{z'_l\}_{l \in [L]}$ recursively as follows:

$$z'_L = z_L,$$

$$z'_l = \begin{cases} z_l, & \text{if } (d_{c_{l+1}} + a_{c_{l+1}}) - (d_{c_l} + a_{c_l}) \ge \phi(z'_{l+1}) - \phi(z_l), \\ \phi^*\Big(\phi(z'_{l+1}) - \big[(d_{c_{l+1}} + a_{c_{l+1}}) - (d_{c_l} + a_{c_l})\big]\Big), & \text{otherwise,} \end{cases} \quad \forall l \in [L-1].$$

Then, we have the following lemma.

Lemma 8. *From the definition of the sequence $\{z'_l\}_{l \in [L]}$, for each $l \in \{1, \ldots, L\}$, we have*

$$d_{c_l} \ge \phi(z'_l).$$

Moreover,

$$\sum_{l=1}^{L} \frac{\alpha}{3} (z_l - z_{l-1}) \Big(2 d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1} \Big)$$

$$\ge \frac{\alpha}{3} \cdot z'_1 \phi(z'_1) + \sum_{l=2}^{L} (z'_l - z'_{l-1}) \Big(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1} \Big). \tag{23}$$

Proof. We prove the two parts of this lemma separately.

Part-I (Proof of $d_{c_l} \geq \phi(z'_l)$). We show $d_{c_l} \geq \phi(z'_l)$ by induction on L. For the base case $L = 1$:

$$z'_1 = z_1,$$

and by the definition of z_1, the probabilistic utilization level of item i at the arrival of request c_1 is at least z_1. Since $x_{c_1} \neq 0$, it follows that

$$d_{c_1} \geq \phi(z_1) = \phi(z'_1).$$

Assume the statement holds for any number of requests in the set $\{c_l\}_{l \in [L]}$ up to $M - 1$. We prove it for $L = M$. By the same argument as in the base case, we have

$$d_{c_M} \geq \phi(z_M) = \phi(z'_M).$$

If $z'_{M-1} = z_{M-1}$, then by the definition of z_{M-1}, we immediately get

$$\phi(z'_{M-1}) \leq d_{c_{M-1}}.$$

Otherwise, suppose

$$(d_{c_M} + a_{c_M}) - (d_{c_{M-1}} + a_{c_{M-1}}) < \phi(z'_M) - \phi(z_{M-1}).$$

From the definition of z'_{M-1}, we have

$$\phi(z'_{M-1}) = \phi(z'_M) - \left[(d_{c_M} + a_{c_M}) - (d_{c_{M-1}} + a_{c_{M-1}}) \right].$$

Since $\phi(z'_M) \leq d_{c_M}$ and $a_{c_{M-1}} < a_{c_M}$ (by the definition of c_l), it follows that

$$\phi(z'_{M-1}) \leq \left[\phi(z'_M) - d_{c_M} \right] + \left[a_{c_{M-1}} - a_{c_M} \right] + d_{c_{M-1}} \leq d_{c_{M-1}}.$$

For the remaining inequalities, for each $l \in \{1, \ldots, M - 2\}$, the induction hypothesis applies to the set of values $\{z_1, \ldots, z_{M-2}\} \cup \{z'_{M-1}\}$, ensuring that $\phi(z'_l) \leq d_{c_l}$ for all $l \in \{1, \ldots, M - 1\}$.

Part-II (Proof of Eq.(23)). We again use induction on L to prove

$$\sum_{l=1}^{L} \frac{\alpha}{3} (z_l - z_{l-1}) \left(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1} \right)$$

$$\geq \frac{\alpha}{3} z'_1 \phi(z'_1) + \sum_{l=2}^{L} (z'_l - z'_{l-1}) \left(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1} \right).$$

For $L = 1$, the statement is trivial since $z_1 = z'_1$. Assume it holds for all sequences $\{z'_l\}$ of length $L - 1$. We prove it for L.

From the definition of z'_{L-1}, we have $z'_{L-1} \geq z_{L-1}$. Thus, we will have the following two cases:

Case 1: $z'_{L-1} > z_{L-1}$. By the induction hypothesis, for the sequence $\{z_1, \ldots, z_{L-2}\} \cup \{z'_{L-1}\}$, we have

$$\sum_{l=1}^{L-2} \frac{\alpha}{3} (z_l - z_{l-1}) \left(2d_{c_l} + a_{c_l} - a_{c_1} \right) + \frac{\alpha}{3} (z'_{L-1} - z_{L-2}) \left(2d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1} \right)$$

$$\geq \frac{\alpha}{3} z'_1 d_{c_1} + \sum_{l=2}^{L-1} (z'_l - z'_{l-1}) \left(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1} \right).$$

39

Next, we add the term $\frac{\alpha}{3}\left(z'_{L-1}-z_{L-2}\right)\left(2d_{c_{L-1}}+a_{c_{L-1}}-a_n\right)$ to both sides of above inequality.

$$\sum_{l=1}^{L-2}\frac{\alpha}{3}(z_l-z_{l-1})\left(2d_{c_l}+a_{c_l}-a_{c_1}\right)+\frac{\alpha}{3}\left(z'_{L-1}-z_{L-2}\right)\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right)+$$
$$\frac{\alpha}{3}\left(z_L-z'_{L-1}\right)\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right)$$
$$\geq\frac{\alpha}{3}\cdot z'_1\cdot d_{c_1}+\sum_{l=2}^{L}(z'_l-z'_{l-1})\cdot\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right).$$

Next, we will upper-bound the left-hand-size of the above inequality as follows:

$$\sum_{l=1}^{L-2}\frac{\alpha}{3}(z_l-z_{l-1})\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right)+\frac{\alpha}{3}\left(z'_{L-1}-z_{L-2}\right)\left(2d_{c_{L-1}}+a_{c_{L-1}}-a_{c_1}-d_{c_1}\right)+$$
$$\frac{\alpha}{3}\left(z_L-z'_{L-1}\right)\left(2d_{c_L}+a_{c_L}-a_{c_1}-d_{c_1}\right))$$
$$\leq\sum_{l=1}^{L-2}\frac{\alpha}{3}(z_l-z_{l-1})\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right)+\frac{\alpha}{3}\left(z_{L-1}-z_{L-2}\right)\left(2d_{c_{L-1}}+a_{c_{L-1}}-a_{c_1}-d_{c_1}\right)+$$
$$\frac{\alpha}{3}\left(z_L-z_{L-1}\right)\left(2d_{c_L}+a_{c_L}-a_{c_1}-d_{c_1}\right))$$
$$=\sum_{l=1}^{L}\frac{\alpha}{3}(z_l-z_{l-1})\left(2d_{c_l}+a_{c_l}-a_{c_1}-d_{c_1}\right),$$

where the inequality follows since $\left(2d_{c_{L-1}}+a_{c_{L-1}}-a_{c_1}-d_{c_1}\right)\leq\left(2d_{c_L}+a_{c_L}-a_{c_1}-d_{c_1}\right)$. Thus in this case the inequality in Eq. (23) follows.

Case 2: $z'_{L-1}=z_{L-1}$. In this scenario, the inequality follows by the same reasoning as previous steps. This completes the proof of Eq. (23). Thus, we complete the proof of the lemma. □

Returning to the point where we left off, we have:

$$\sum_{j\in C_n^{(i)}}\Delta_{\Lambda_n^{(i)}}^{(j)}\geq\sum_{l=1}^{L}\frac{\alpha}{3}(z_l-z_{l-1})\left(d_{c_l}+a_{c_l}+d_{c_l}-a_{c_1}-d_{c_1}\right)$$

$$\geq\frac{\alpha}{3}\left[z'_1\phi(z'_1)+\sum_{l=2}^{L}(z'_l-z'_{l-1})\left(d_{c_l}+a_{c_l}+d_{c_l}-a_{c_1}-d_{c_1}\right)\right]$$

$$\geq\frac{\alpha}{3}\left[z'_1\phi(z'_1)+\sum_{l=2}^{L}(z'_l-z'_{l-1})\left(2\phi(z'_l)-\phi(z'_1)\right)\right]$$

$$\geq\frac{\alpha}{3}\left[z'_1\phi(z'_1)+\int_{\eta=z'_1}^{z'_L}\left(2\phi(\eta)-\phi(z'_1)\right)d\eta\right]$$

$$=\frac{2\alpha}{3}z'_1\phi(z'_1)-\frac{\alpha}{3}z'_L\phi(z'_1)+\int_{z'_1}^{z'_L}\frac{2\alpha}{3}\phi(\eta)d\eta,$$

where the second and third inequalities follow from Lemma 8, and the fourth inequality holds because ϕ is an increasing function.

Next, we upper-bound $\sum_{j \in B_n^{(i)} \setminus C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)}$ as follows:

$$\sum_{j \in B_n^{(i)} \setminus C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)} \geq \frac{\alpha}{3} d_n \max\left\{0, y_n^{(i_n^*)} - z_L'\right\}.$$

The above inequality holds because, for each request j in $B_n^{(i)} \setminus C_n^{(i)}$, we have $a_j + d_j > a_n + d_n$ by the definitions of $B_n^{(i)}$ and $C_n^{(i)}$. Then, from Eqs. (20)–(21), for each such j, $\Delta_{\Lambda_n^{(i)}}^{(j)}$ is at least $\frac{\alpha}{3} d_n \hat{x}_j$. Furthermore,

$$\sum_{j \in B_n^{(i)} \setminus C_n^{(i)}} \hat{x}_j \geq \max\left\{0, y_n^{(i_n^*)} - z_L'\right\}.$$

We move on to lower-bound the term u_n. From the update rule in Eq. (19), for each request n we have:

$$u_n \geq \frac{\alpha}{3} d_n \hat{x}_n$$

$$\geq \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - y_n^{(i_n^*)}\right\},$$

where the second inequality follows from the way Algorithm 4 sets the value of $\hat{x}_n$ in Eq. (6).

Combining bounds obtained for the LHS of Eq. (18), we obtain

$$u_n + \sum_{j=n}^{N} \lambda_j^{(i)} \mathbb{I}_{\{a_n + d_n > a_j\}} \geq \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - y_n^{(i_n^*)}\right\} + \sum_{j \in B_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)}$$

$$= \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - y_n^{(i_n^*)}\right\} + \sum_{j \in C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)} + \sum_{j \in B_n^{(i)} \setminus C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)}$$

$$\geq \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - y_n^{(i_n^*)}\right\} + \frac{2\alpha}{3} z_1' \phi(z_1') - \frac{\alpha}{3} z_L' \phi(z_1')$$

$$+ \int_{z_1'}^{z_L'} \frac{2\alpha}{3} \phi(\eta) d\eta + \frac{\alpha}{3} d_n \max\left\{0, y_n^{(i_n^*)} - z_L'\right\}$$

$$= \frac{2\alpha}{3} z_1' \phi(z_1') - \frac{\alpha}{3} z_L' \phi(z_1') + \int_{z_1'}^{z_L'} \frac{2\alpha}{3} \phi(\eta) d\eta$$

$$+ \frac{\alpha}{3} d_n \max\left\{0, y_n^{(i_n^*)} - z_L'\right\} + \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - y_n^{(i_n^*)}\right\}.$$

$$\geq \frac{2\alpha}{3} z_1' \phi(z_1') - \frac{\alpha}{3} z_L' \phi(z_1') + \int_{z_1'}^{z_L'} \frac{2\alpha}{3} \phi(\eta) d\eta$$

$$+ \frac{\alpha}{3} d_n \max\left\{0, \phi^*(d_n) - z_L'\right\}$$

$$\geq d_n,$$

where the last inequality follows from the lemma given below. Consequently, we conclude that the second set of constraints in Eq. (18) is satisfied by the design of system variables in Eqs. (19)–(21). Therefore, **if the increasing pricing function ϕ satisfies the system of constraints in Theorem 4, it establishes the α-competitiveness of Algorithm 4. Thus, to complete the proof of Theorem 4, it suffices to prove the following lemma.**

Lemma 9. *If the ϕ function satisfy the constraints in Eqs. (7), then*

$$\frac{2\alpha}{3}z_1'\phi(z_1') - \frac{\alpha}{3}z_L'\phi(z_1') + \int_{z_1'}^{z_L'} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\left\{0, \phi^*(d_n) - z_L'\right\} \geq d_n,$$

$$\forall d_n \in [d_{\min}, d_{\max}], z_1' \in \left(0, z_L'\right], z_L' \in \left(0, \phi^*(d_{min})\right].$$

Proof. Let us denote the left-hand side of the above inequality by

$$\mathfrak{L}(z_1', z_L', d_n),$$

which is a function of z_1', z_L', and d_n. We are going to prove that

$$\mathfrak{L}(z_1', z_L', d_n) \geq \min\left\{\mathfrak{L}\left(\frac{z_L'}{2}, z_L', d_n\right), \quad \mathfrak{L}(z_L', z_L', d_n)\right\}.$$

Step 1: Find the critical point in the interior. Fix z_L' and d_n and differentiate L with respect to z_1'. A straightforward calculation shows

$$\frac{\partial \mathfrak{L}}{\partial z_1'} = \frac{\alpha}{3}\phi'(z_1')\left(2z_1' - z_L'\right).$$

Since $\phi'(z_1') \neq 0$, setting this derivative to zero forces

$$z_1' = \frac{z_L'}{2}.$$

A direct substitution $z_1' = \frac{z_L'}{2}$ simplifies the first two terms and yields

$$\mathfrak{L}\left(\frac{z_L'}{2}, z_L', d_n\right) = \int_{z_L'/2}^{z_L'} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\{0, \phi^*(d_n) - z_L'\}.$$

Step 2: Evaluate $\mathfrak{L}$ on the boundary plane where $z_1' = z_L'$. When $z_1' = z_L'$, the integral term vanishes. One obtains

$$\mathfrak{L}(z_L', z_L', d_n) = \frac{\alpha}{3}z_L'\phi(z_L') + \frac{\alpha}{3}d_n \max\{0, \phi^*(d_n) - z_L'\}.$$

This can be written as

$$\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}z_L'\left(d_n - \phi(z_L')\right)$$

for values of $z_L' \leq \phi^*(d_n)$.

Step 3: Evaluate $\mathfrak{L}$ on the boundary plane where z_1' converges to zero. It can be verified that $\mathfrak{L}(z_1', z_L', d_n)$ is lower-bounded by $L(z_L'/2, z_L', d_n)$, as z_1' converges to zero, as follows:

$$\mathfrak{L}(0, z_L', d_n) = \int_0^{z_L'} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\left\{0, \phi^*(d_n) - z_L'\right\} - \frac{\alpha}{3}z_L'\phi(z_1')$$

$$\geq \int_{z_L'/2}^{z_L'} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\left\{0, \phi^*(d_n) - z_L'\right\} = L(z_L'/2, z_L', d_n),$$

where the inequality follows from the fact that ϕ function is increasing.

Putting together the above results, in the above three steps, since a continuous function on a closed interval attains its minimum either at a critical point or on the boundary, for each fixed z'_L and d_n we have

$$\mathfrak{L}(z'_1, z'_L, d_n) \geq \min\left\{\mathfrak{L}\left(\frac{z'_L}{2}, z'_L, d_n\right), \quad \mathfrak{L}(z'_L, z'_L, d_n)\right\}$$

$$= \min\left\{\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}z'_L\left(d_n - \phi(z'_L)\right),\right.$$

$$\left.\int_{z'_L/2}^{z'_L}\frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\max\{0, \phi^*(d_n) - z'_L\}\right\}$$

$$\geq d_n,$$

where the last inequality follows from the constraints in Eqs. (7). This completes the proof. □

G   Proof of Order-Optimality of Algorithm 4

In the following corollary, we present an analytical design of ϕ that satisfies the system of constraints in Theorem 4. Although this design is not the optimal design that satisfies the set of constraints in Theorem 4, it is sufficient to obtain the order optimality of Algorithm 4.

Corollary 1. *DOP-ϕ-VARIABLE is* $3 \cdot \left(1 + \ln\left(\frac{d_{\max}}{d_{\min}}\right)\right)$*-competitive for the kRental-Variable problem if the pricing function ϕ is designed as:*

$$\phi(y) = d_{\min} \cdot \exp\left(\left[1 + \ln\left(\frac{d_{\max}}{d_{\min}}\right)\right] \cdot y - 1\right), \quad \forall y \in [0, 1].$$

Proof. We now prove that for the given design of the ϕ function, the constraints in Theorem 4 (see Eqs. (7)) are satisfied when

$$\alpha = 3 \cdot \left(1 + \ln\frac{d_{\max}}{d_{\min}}\right).$$

Proof for the first inequality of Eq. (7). The left-hand side of Eq. (7) can be lower-bounded as follows:

$$\int_{\eta=y_1}^{2y_1}\frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\left(\phi^*(d_n) - 2y_1\right)$$

$$\geq \int_{\eta=0}^{2y_1}\frac{\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\left(\phi^*(d_n) - 2y_1\right)$$

$$\geq \int_{\eta=0}^{\phi^*(d_n)}\frac{\alpha}{3}\phi(\eta)d\eta$$

$$\geq d_n,$$

where the first and second inequality follows because ϕ is an increasing function. The final inequality is a consequence of the specific design of the ϕ function stated in the corollary.

43

Proof for the second inequality of Eq. (7). The left-hand side of the second inequality of Eq. (7) can be lower-bounded as:

$$\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}y_2\Big(d_n - \phi(y_2)\Big)$$

$$=\frac{\alpha}{3}y_2\phi(y_2) + \frac{\alpha}{3}d_n\Big(\phi^*(d_n) - y_2\Big)$$

$$\geq \int_{\eta=0}^{\phi^*(d_n)} \frac{\alpha}{3}\phi(\eta)d\eta$$

$$\geq d_n,$$

where the equality is obtained by rearranging the terms. The first inequality follows from the increasing nature of ϕ and the corresponding definitions of y_2 and $\phi^*(d_n)$. The final inequality is ensured by the design of the ϕ function provided in the corollary.

Since the proposed design for the ϕ function satisfies both sets of constraints, it follows that Algorithm 4 obtains the competitive ratio of $3 \cdot \left(1 + \ln \frac{d_{\max}}{d_{\min}}\right)$ for the given design in the corollary. □

Next, we prove the following lower bound on the competitiveness of any online algorithm for the kRental-Variable problem.

Proposition 5 (Lower Bound of kRental-Variable). *Under Assumption 2, no online algorithm, deterministic or randomized, can obtain a competitive ratio better than $1 + \ln\left(\frac{d_{\max}}{d_{\min}}\right)$ for the kRental-Variable problem.*

Proof. Following the same proof structure as in [29] for the online selection problem, we can prove the lower bound $1+\ln\left(\frac{d_{\max}}{d_{\min}}\right)$ on the competitiveness of every online algorithm for the kRental-Variable problem. Based on Corollary 1, we can then claim that Algorithm 4 is order-optimal.

We design a set of hard instances for the kRental-Variable problem similar to [29]. Let $\mathcal{A}(k, d)$ denote a batch of k identical requests, each with a rental duration request d (with $d \in [d_{\min}, d_{\max}]$). Divide the uncertainty range $[d_{\min}, d_{\max}]$ into $m - 1$ sub-ranges of equal length

$$\Delta_d = \frac{d_{\max} - d_{\min}}{m - 1}.$$

Let $\mathcal{D} := \{d_i\}_{i\in[m]}$, where $d_i = d_{\min} + (i - 1)\Delta_d$ for $i \in [m]$. Define an instance

$$I_{d_i} := \mathcal{A}(k, d_1) \oplus \mathcal{A}(k, d_2) \oplus \cdots \oplus \mathcal{A}(k, d_i),$$

which consists of a sequence of request batches with increasing rental duration requests, arriving in an arbitrarily short interval one after the other (here, $\mathcal{A}(k, d_i) \oplus \mathcal{A}(k, d_j)$ denotes a batch $\mathcal{A}(k, d_i)$ followed by a batch $\mathcal{A}(k, d_j)$). In this construction, all requests—from the first batch with valuation d_1 to the last batch with valuation d_m—arrive within a short time interval $[0, \epsilon]$, where ϵ is a small constant satisfying $\epsilon < d_{\min}$.

We consider the collection $\{I_{d_i}\}_{i\in[m]}$ as a family of hard instances for the kRental-Variable problem. Using the same proof structure as those of [29], we derive the optimal competitive ratio that any online algorithm can achieve on these hard instances. Consequently, we establish the lower bound $1 + \ln\left(\frac{d_{\max}}{d_{\min}}\right)$ for the competitive ratio of every online algorithm in this setting. □

By the above lower bound result and Corollary 1, we argue that DOP-ϕ-VARIABLE is order-optimal.

H   Numerical Computation of ϕ Satisfying Constraints in Theorem 4

To numerically obtain a ϕ function that satisfies the constraints in Theorem 4 with lowest value of α, we discretize the interval $[0, 1]$ using a parameter $\epsilon \in (0, 1)$, allowing the function ϕ to change only at the points $\{\epsilon \cdot i\}_{i=1}^{\lceil 1/\epsilon \rceil}$.

By doing so, we transform the continuous constraints of Theorem 4 into a finite system of constraints, leading to an optimization problem with a finite number of variables and constraints. The optimal solution for the set of variables $\{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}$ in the following discretized LP provides the desired price function design ϕ; specifically, the values of ϕ at the points $\{\epsilon \cdot i\}_{i=1}^{\lceil 1/\epsilon \rceil}$ are defined according to the solution $\{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}$.

Proposition 6. *For any given $\epsilon \in (0, 1)$, DOP-ϕ-VARIABLE is α^*-competitive for the kRental-Variable problem if the pricing function ϕ is designed as*

$$\phi(y) = \pi_{\lceil \frac{y}{\epsilon} \rceil}^{(\epsilon)}, \quad \forall y \in [0, 1],$$

where the set of prices $\{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}$ and the competitive ratio parameter α^* are the optimal solution to the following LP[1]:

$$\alpha^* = \min_{\alpha, \ \{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}} \alpha \tag{24a}$$

$$\text{s.t.} \quad \pi_i^{(\epsilon)} \leq \pi_{i+1}^{(\epsilon)}, \quad \forall i \in \left\{1, 2, \ldots, \lceil \tfrac{1}{\epsilon} \rceil - 1\right\}, \tag{24b}$$

$$\pi_1^{(\epsilon)} = d_{\min}, \tag{24c}$$

$$\pi_{\lceil \frac{1}{\epsilon} \rceil}^{(\epsilon)} \geq d_{\max}, \tag{24d}$$

$$\sum_{j=i}^{2i} \frac{2\alpha}{3}\left(\pi_{j-1}^{(\epsilon)}\right)\epsilon + \frac{\alpha}{3}\left(\epsilon \cdot l - 2\epsilon \cdot (i+1)\right)\pi_{l-1}^{(\epsilon)} \geq \pi_l^{(\epsilon)},$$

$$\forall i \in \left\{1, \ldots, \lfloor \tfrac{l}{2} \rfloor\right\} \tag{24e}$$

$$\frac{\alpha}{3}\pi_{l-1}^{(\epsilon)}\epsilon l - \frac{\alpha}{3}\left(\pi_l^{(\epsilon)} - \pi_{i+1}^{(\epsilon)}\right)\epsilon \cdot (i+1) \geq \pi_l^{(\epsilon)},$$

$$\forall i \in \{1, \ldots, l\}. \tag{24f}$$

Proof. We need to prove that for each $d_n \in [d_{\min}, d_{\max}]$, for all

$$y_1 \in \left[0, \frac{\phi^*(d_n)}{2}\right] \quad \text{and} \quad y_2 \in \left[0, \phi^*(d_n)\right],$$

the following constraints for ϕ are satisfied:

$$\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\left(\phi^*(d_n) - 2y_1\right) \geq d_n, \tag{25}$$

$$\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}y_2\left(d_n - \phi(y_2)\right) \geq d_n. \tag{26}$$

where $\phi^*(d_n) = \sup\left\{x \in [0, 1] \mid \phi(x) \leq d_n\right\}$.

---

[1] Note that this LP can be solved by standard LP solvers. In addition, it is worth noting that the number of variables in the LP grows on the order of $\lceil 1/\epsilon \rceil$, while the number of constraints increases on the order of $(\lceil 1/\epsilon \rceil)^2$. Standard techniques such as the interior point method can be employed to solve this optimization problem efficiently.

For any value of $d_n \in [d_{\min}, d_{\max}]$, following the design of the ϕ function, whose value changes only at the points

$$\left\{ \epsilon, 2\epsilon, \ldots, \lceil \tfrac{1}{\epsilon} \rceil \cdot \epsilon \right\},$$

we have $\phi^*(d_n) = \epsilon \cdot m$ for some $m \in \left\{ 1, 2, \ldots, \lceil \tfrac{1}{\epsilon} \rceil \right\}$ because of the constraints in Eq. (24c) and Eq. (24d).

Thus, for any $y_1 \le \frac{\phi^*(d_n)}{2} = \frac{\epsilon \cdot m}{2}$, the left-hand side of the constraint in Eq. (25) can be lower-bounded as follows:

$$\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3} \phi(\eta) d\eta + \frac{\alpha}{3} d_n \Big(\phi^*(d_n) - 2y_1 \Big)$$

$$= \int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3} \phi(\eta) d\eta + \frac{\alpha}{3} d_n \Big(\epsilon \cdot m - 2y_1 \Big)$$

$$\ge \sum_{j=\lfloor \frac{y_1}{\epsilon} \rfloor \cdot \epsilon}^{2 \lfloor \frac{y_1}{\epsilon} \rfloor \cdot \epsilon} \frac{2\alpha}{3} \Big(\phi\big(\epsilon \cdot j\big) - \phi\big(\epsilon \cdot (j-1)\big) \Big) \epsilon + \frac{\alpha}{3} \phi\big(\epsilon \cdot (m-1)\big) \Big(\epsilon \cdot m - 2\epsilon \cdot \big(\lfloor \tfrac{y_1}{\epsilon} \rfloor + 1 \big) \Big)$$

$$= \sum_{j=i \cdot \epsilon}^{2i \cdot \epsilon} \frac{2\alpha}{3} \Big(\phi\big(\epsilon \cdot j\big) - \phi\big(\epsilon \cdot (j-1)\big) \Big) \epsilon + \frac{\alpha}{3} \phi\big(\epsilon \cdot (m-1)\big) \Big(\epsilon \cdot m - 2\epsilon \cdot (i+1) \Big)$$

$$\ge \phi\big(\epsilon \cdot m\big) \ge d_n,$$

where $i = \lfloor \frac{y_1}{\epsilon} \rfloor$. The second inequality follows from some simple algebra, the third inequality follows from the constraint in Eq. (24e) (since $i \le \frac{m}{2}$), and the last inequality follows from the definition of ϕ^* and m.

Thus, the first constraint in Eq. (25) is satisfied for every $d_n \in [d_{\min}, d_{\max}]$ and every $y_1 \in \left[0, \frac{\phi^*(d_n)}{2} \right]$.

Next, let us prove the second inequality in Eq. (7). The LHS of that constraint can be lower-bounded as follows:

$$\frac{\alpha}{3} d_n \phi^*(d_n) - \frac{\alpha}{3} y_2 \Big(d_n - \phi(y_2) \Big)$$

$$\ge \frac{\alpha}{3} \phi\big(\epsilon \cdot (m-1)\big) \epsilon \cdot m - \frac{\alpha}{3} \epsilon \cdot \Big(\lfloor \tfrac{y_2}{\epsilon} \rfloor + 1 \Big) \Big(\phi\big(\epsilon \cdot m\big) - \phi\big(\lfloor \tfrac{y_2}{\epsilon} \rfloor \big) \Big)$$

$$= \frac{\alpha}{3} \phi\big(\epsilon \cdot (m-1)\big) \epsilon \cdot m - \frac{\alpha}{3} \epsilon \cdot (i+1) \Big(\phi\big(\epsilon \cdot m\big) - \phi\big((i+1) \cdot \epsilon\big) \Big)$$

$$\ge \phi\big(\epsilon \cdot m\big) \ge d_n,$$

where $i = \lfloor \frac{y_2}{\epsilon} \rfloor$, the first inequality follows from simple algebra, the second inequality follows from the constraint in Eq. (24f) (since $y_2 \in [0, \phi^*(d_n)]$, thus $i \le m$), and the final inequality follows from the definition of the ϕ function. Thus, the constraint in Eq. (26) is satisfied for every $d_n \in [d_{\min}, d_{\max}]$ and for every $y_2 \in \left[0, \phi^*(d_n) \right]$. We thus complete the proof. $\qquad \square$

I   Fractional kRental-Variable Problem

In this section, we study the kRental-Variable problem in the fractional setting, where the decision maker can allocate a fractional portion of each unit of resource to a request. We will compete against the optimal

clairvoyant algorithm, whose objective value $\mathsf{OPT}(I)$ can be computed as follows:

$$\max_{\mathbf{x}} \quad \sum_{n \in [N]} d_n \cdot x_n, \tag{27}$$

$$s.t. \quad \sum_{j \in [n]} \hat{x}_j \cdot 1_{\{d_j + a_j > a_n\}} \leq k, \quad \forall n \in [N], \tag{28}$$

$$x_n \in [0, 1], \quad \forall n \in [N]. \tag{29}$$

We present DOP-ϕ-VARIABLE-FRACTIONAL in Algorithm 5, which also adopts a utility-maximizing rule similar to Algorithm 3 to produce fractional allocations; the key difference is that the decision for each request now depends on the *total* utilization level of the entire item inventory, rather than on the state of a single unit.

Algorithm 5: Duration Oblivious Price-based Algorithm for Fractional kRental-Variable (DOP-ϕ-VARIABLE-FRACTIONAL)

---

1 **Input:** Pricing function $\phi : [0, 1] \rightarrow [d_{min}, d_{max}]$
2 **while** *a new request n arrives* **do**
3   Observe the request's duration request, d_n.
4   Compute utilization level of the inventory at the arrival of request n, where

$$y_n = \sum_{j=1}^{n-1} \hat{x}_j \cdot \mathbb{I}_{\{a_j + d_j > a_n\}} \tag{30}$$

5   Compute the fractional allocation $\hat{x}_n$ as follows

$$\hat{x}_n = \underset{x \in \left[0, \min\{1, k - y_n\}\right]}{\arg\max} \left\{ x \cdot d_n - k \int_{\eta = y_n/k}^{(y_n + x)/k} \phi(\eta) \, d\eta \right\}. \tag{31}$$

6   Allocate the partial allocation $\hat{x}_n$ to request n.
7 **end**

Proposition 7. *DOP-ϕ-VARIABLE-FRACTIONAL is α-competitive for the fractional kRental-Variable problem, provided that the price function ϕ is increasing and is designed to satisfy the following inequalities (parameterized by α) for all $d_n \in \left[d_{\min}, d_{\max}\right]$ and $n \in [N]$:*

$$\int_{\eta = y_1}^{2y_1} \alpha \, \phi(\eta) \, d\eta + \frac{\alpha}{2} \, d_n \left(\phi^*(d_n) - 2 \, y_1 \right) \geq d_n, \quad \forall \, y_1 \in \left[0, \frac{\phi^*(d_n)}{2}\right], \tag{32}$$

$$\frac{\alpha}{2} \, d_n \, \phi^*(d_n) - \frac{\alpha}{2} \, y_2 \left(d_n - \phi(y_2) \right) \geq d_n, \quad \forall \, y_2 \in \left[0, \, \phi^*(d_n)\right]. \tag{33}$$

where $\phi^*(d_n) = \sup_{x \in [0,1]}\{\phi(x) \leq d_n\}$.

The proof of the above proposition is given in the following. As with Theorem 4, obtaining a closed-form expression for the pricing function ϕ that minimizes α while satisfying the constraints in the above proposition is challenging, as it requires solving a system of delayed differential inequalities involving an inverse term. Therefore, we adopt the same numerical approach used to construct a valid pricing function ϕ in Proposition 6. Specifically, we discretize the interval $[0, 1]$ using a parameter $\epsilon \in (0, 1)$, allowing ϕ to change only at the points $\{\epsilon \cdot i\}_{i=1}^{\lceil 1/\epsilon \rceil}$. Using this discretization, we obtain a pricing function ϕ that satisfies

the constraints of Proposition 7. The competitive ratio of Algorithm 5 using this numerically constructed pricing function as input is shown in Figure 2.

Proof of Proposition 7. The proof structure is similar to the proof of Theorem 4. Specifically, we still adopt an online primal-dual approach to establish the competitive ratio of Algorithm 5. Consider the dual LP corresponding to the primal LP in Eq. (5):

$$\min_{\lambda, \mathbf{u}} \quad \sum_{n \in [N]} u_n + k \cdot \sum_{n \in [N]} \lambda_n, \tag{34}$$

$$\text{s.t.} \quad u_n + \sum_{j=n}^{N} \lambda_j \cdot \mathbb{I}_{\{a_n + d_n > a_j\}} \geq d_n, \quad \forall n \in [N], \tag{35}$$

$$\lambda_n \geq 0, \quad \forall n \in [N].$$

Let $\mathsf{ALG}(\mathcal{I}) = \sum_{n \in [N]} d_n \cdot \hat{x}_n$ represent the expected objective value of Algorithm 5 on an instance $\mathcal{I}$.

In the first step, we design a feasible solution for the dual problem, denoted as $\{u_n, \lambda_n\}_{n \in [N]}$, which corresponds to the dual linear program, and define the dual objective

$$D = \sum_{n \in [N]} u_n + k \cdot \sum_{n \in [N]} \lambda_n.$$

We first establish that for the designed dual solution we have $D = \alpha \cdot \mathsf{ALG}(\mathcal{I})$ for every instance $\mathcal{I}$ of the problem. We then establish that the dual vector $\{u_n, \lambda_n\}_{n \in [N]}$ is feasible for the dual LP in Eq. (34). Thus, the α-competitiveness of Algorithm 5 follows. Let us initialize all dual variables to zero. For each rental request n, let $\hat{x}_n$ be the fractional allocation determined by Algorithm 5. We then perform the following updates:

$$u_n = \frac{\alpha}{2} \left(d_n - \phi\left((y_n + \hat{x}_n)/k\right) \right) \hat{x}_n, \tag{36}$$

$$\lambda_j \leftarrow \lambda_j + \frac{\alpha}{2} \left(a_{j+1} - a_j \right) \frac{\hat{x}_n}{k}, \quad \forall j, n \leq j < v_n^*, a_j < a_n + d_n, \tag{37}$$

$$\lambda_{v_n^*} \leftarrow \lambda_{v_n^*} + \frac{\alpha}{2} \left(d_n - (a_{v_n^*} - a_n) + \phi\left((y_n + \hat{x}_n)/k\right) \right) \frac{\hat{x}_n}{k}, \tag{38}$$

where $v_n^* = \max\{ j \geq n : a_j < a_n + d_n \}$.

Next, let Δ_n^D denote the increase in the dual solution objective value (Eq. (34)) after updating the variables of the system according to Eqs. (36)–(6) for the n-th rental request. It follows directly that

$$\Delta_n^D = \alpha \, d_n \hat{x}_n.$$

Summing over all requests $n \in [N]$, we have $\alpha \cdot \mathsf{ALG}(\mathcal{I}) = \alpha \cdot \sum_{n \in [N]} d_n \cdot \hat{x}_n = \alpha \cdot D$.

Feasibility of the dual constraints in Eq. (35). For each request n in instance $\mathcal{I}$, we show that

$$\sum_{j=n}^{N} \lambda_j \cdot \mathbb{I}_{\{a_n + d_n > a_j\}} + u_n \geq d_n.$$

Let us define

$$\Lambda_n = \sum_{j=n}^{N} \lambda_j \cdot \mathbb{I}_{\{a_n + d_n > a_j\}}.$$

Let us further define the set of requests B_n by

$$B_n = \left\{ 1 \le j < n \mid a_j + d_j > a_n, \hat{x}_j > 0 \right\}. \tag{39}$$

In other words, B_n is the set of all requests who arrive before request n, request rental intervals overlapping the interval of request n, and have a non-zero probability of receiving unit i.

Furthermore, define $C_n \subseteq B_n$ by

$$C_n = \left\{ j \in B_n \mid a_j + d_j < a_n + d_n \right\}.$$

Next, let the set of requests $\{c_l\}_{l \in [L]}$ be defined recursively such that

$$c_1 = \underset{j \in C_n}{\arg\min}\{a_j + d_j\},$$

$$c_l = \underset{\substack{j \in C_n \\ a_j > a_{c_{l-1}}}}{\arg\min}\{a_j + d_j\}, \qquad 1 < l \le L.$$

The request $c_L \in C_n$ is the one for which $\{j \in C_n \mid a_j > a_{c_L}\}$ is empty. Using the sequence of requests $\{c_l\}_{l \in [L]}$, we partition C_n into L sets:

$$C_l = \left\{ j \in C_n \mid a_{c_{l-1}} \le a_j < a_{c_l} \right\}, \quad \forall l \in [L].$$

Furthermore, define the set of values $\{z_l\}_{l \in [L]}$ by

$$z_l = \sum_{\substack{j \in C_n \\ a_j < a_{c_l}}} \hat{x}_j / k.$$

Let $\Delta_{\Lambda_n}^{(j)}$ denote the increase in Λ_n after processing the rental request of request j and updating the variables according to Eqs. (37)–(38). From the way we perform the updates, it follows that

$$\sum_{j \in C_n} \Delta_{\Lambda_n}^{(j)} \ge \sum_{l=1}^{L} \sum_{j \in C_l} \frac{\alpha}{2} \frac{\hat{x}_j}{k} \left(\phi\big((y_j + \hat{x}_j)/k\big) + a_j + d_j - a_n \right)$$

$$\ge \sum_{l=1}^{L} \sum_{j \in C_l} \frac{\alpha}{2} \frac{\hat{x}_j}{k} \left(a_j + d_j - a_n \right) + \frac{\alpha}{2} \int_0^{z_L} \phi(\eta)\, d\eta$$

$$\ge \sum_{l=1}^{L} \sum_{j \in C_l} \frac{\alpha}{2} \frac{\hat{x}_j}{k} \left(a_{c_l} + d_{c_l} - a_n \right) + \frac{\alpha}{2} \int_0^{z_L} \phi(\eta)\, d\eta$$

$$\ge \sum_{l=1}^{L} \frac{\alpha}{2} \left(z_l - z_{l-1} \right) \left(d_{c_l} + a_{c_l} - a_{c_1} \right) + \frac{\alpha}{2} \int_0^{z_L} \phi(\eta)\, d\eta. \tag{40}$$

where the first inequality follows from the dual updates in Eqs. (37)–(38), the second inequality follows from rearranging the terms and the fact that ϕ function is increasing. The third inequality relies on the definition of c_l and C_l (under which $d_j \le d_{c_l}$ and $a_j + d_j \ge a_{c_l} + d_{c_l}$ for each $j \in C_l$), and the last inequality follows from the definition of the values $\{z_l\}_{l \in [L]}$, taking $z_0 = 0$.

Define the sequence $\{z_l'\}_{l \in [L]}$ recursively as follows:

$$z_L' = z_L,$$

$$z_l' = \begin{cases} z_l, & \text{if } \left(d_{c_{l+1}} + a_{c_{l+1}} \right) - \left(d_{c_l} + a_{c_l} \right) \ge \phi(z_{l+1}') - \phi(z_l), \\ \phi^*\Big(\phi(z_{l+1}') - \big[\left(d_{c_{l+1}} + a_{c_{l+1}} \right) - \left(d_{c_l} + a_{c_l} \right) \big] \Big), & \text{otherwise,} \end{cases} \qquad \forall l \in [L-1].$$

49

Lemma 10. *From the definition of the sequence $\{z'_l\}_{l \in [L]}$, for each $l \in \{1, \ldots, L\}$, we have*

$$d_{c_l} \geq \phi(z'_l).$$

Moreover,

$$\sum_{l=1}^{L} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right)$$

$$\geq \frac{\alpha}{2} \cdot z'_1 \phi(z'_1) + \sum_{l=2}^{L}(z'_l - z'_{l-1})\left(a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\right). \tag{41}$$

Proof. Again, we prove the two parts of the lemma separately.

Part-I (Proof of $d_{c_l} \geq \phi(z'_l)$). We show $d_{c_l} \geq \phi(z'_l)$ by induction on L. For the base case $L = 1$:

$$z'_1 = z_1,$$

and by the definition of z_1, the probabilistic utilization level of item i at the arrival of request c_1 is at least z_1. Since $x_{c_1} \neq 0$, it follows that

$$d_{c_1} \geq \phi(z_1) = \phi(z'_1).$$

Assume the statement holds for any number of requests in the set $\{c_l\}_{l \in [L]}$ up to $M - 1$. We prove it for $L = M$. By the same argument as in the base case, we have

$$d_{c_M} \geq \phi(z_M) = \phi(z'_M).$$

If $z'_{M-1} = z_{M-1}$, then by the definition of z_{M-1}, we immediately get

$$\phi(z'_{M-1}) \leq d_{c_{M-1}}.$$

Otherwise, suppose

$$\left(d_{c_M} + a_{c_M}\right) - \left(d_{c_{M-1}} + a_{c_{M-1}}\right) < \phi(z'_M) - \phi(z_{M-1}).$$

From the definition of z'_{M-1}, we have

$$\phi(z'_{M-1}) = \phi(z'_M) - \left[\left(d_{c_M} + a_{c_M}\right) - \left(d_{c_{M-1}} + a_{c_{M-1}}\right)\right].$$

Since $\phi(z'_M) \leq d_{c_M}$ and $a_{c_{M-1}} < a_{c_M}$ (by the definition of c_l), it follows that

$$\phi(z'_{M-1}) \leq \left[\phi(z'_M) - d_{c_M}\right] + \left[a_{c_{M-1}} - a_{c_M}\right] + d_{c_{M-1}} \leq d_{c_{M-1}}.$$

For the remaining inequalities, for each $l \in \{1, \ldots, M - 2\}$, the induction hypothesis applies to the set of values $\{z_1, \ldots, z_{M-2}\} \cup \{z'_{M-1}\}$, ensuring that $\phi(z'_l) \leq d_{c_l}$ for all $l \in \{1, \ldots, M - 1\}$.

Part-II (Proof of (41)). We again use induction on L to prove

$$\sum_{l=1}^{L} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right)$$

$$\geq \frac{\alpha}{2} z'_1 \phi(z'_1) + \sum_{l=2}^{L}(z'_l - z'_{l-1})\left(a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\right).$$

For $L = 1$, the statement is trivial since $z_1 = z_1'$. Assume it holds for all sequences $\{z_l'\}$ of length $L - 1$. We prove it for L.

From the definition of z_{L-1}', we have $z_{L-1}' \geq z_{L-1}$. Thus, we will have the following two cases:

Case 1: $z_{L-1}' > z_{L-1}$. By the induction hypothesis, for the sequence $\{z_1, \ldots, z_{L-2}\} \cup \{z_{L-1}'\}$, we have

$$\sum_{l=1}^{L-2} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1}\right) + \frac{\alpha}{2}(z_{L-1}' - z_{L-2})\left(d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\right)$$

$$\geq \frac{\alpha}{2}z_1' d_{c_1} + \sum_{l=2}^{L-1}(z_l' - z_{l-1}')\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right).$$

Next, we add the term $\frac{\alpha}{3}(z_{L-1}' - z_{L-2})\left(d_{c_{L-1}} + a_{c_{L-1}} - a_n\right)$ to both sides of above inequality.

$$\sum_{l=1}^{L-2} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1}\right) + \frac{\alpha}{2}(z_{L-1}' - z_{L-2})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right) +$$

$$\frac{\alpha}{2}(z_L - z_{L-1}')\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right)$$

$$\geq \frac{\alpha}{2} \cdot z_1' \cdot d_{c_1} + \sum_{l=2}^{L}(z_l' - z_{l-1}') \cdot \left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right).$$

Next, we will upper-bound the left-hand-size of the above inequality as follows:

$$\sum_{l=1}^{L-2} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right) + \frac{\alpha}{2}(z_{L-1}' - z_{L-2})\left(d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\right) +$$

$$\frac{\alpha}{2}(z_L - z_{L-1}')\left(d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\right))$$

$$\leq \sum_{l=1}^{L-2} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right) + \frac{\alpha}{2}(z_{L-1} - z_{L-2})\left(d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\right) +$$

$$\frac{\alpha}{2}(z_L - z_{L-1})\left(d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\right))$$

$$= \sum_{l=1}^{L} \frac{\alpha}{2}(z_l - z_{l-1})\left(d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\right),$$

where the inequality follows since $\left(d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\right) \leq \left(d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\right)$. Thus in this case the inequality in Eq. (41) follows.

Case 2: $z_{L-1}' = z_{L-1}$. In this scenario, the inequality follows by the same reasoning as previous steps.

This completes the proof of (41). $\qquad\square$

Returning to the point where we left off, we have:

$$\sum_{j \in C_n^{(i)}} \Delta_{\Lambda_n^{(i)}}^{(j)} \geq \sum_{l=1}^{L} \frac{\alpha}{2}(z_l - z_{l-1})(a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}) + \frac{\alpha}{2} \int_{\eta=0}^{z_L'} \phi(\eta)d\eta$$

$$\geq \frac{\alpha}{2}\left[z_1'\phi(z_1') + \sum_{l=2}^{L}(z_l' - z_{l-1}')(a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1})\right] + \frac{\alpha}{2} \int_{\eta=0}^{z_L'} \phi(\eta)d\eta$$

$$\geq \frac{\alpha}{2}\left[z_1'\phi(z_1') + \sum_{l=2}^{L}(z_l' - z_{l-1}')(\phi(z_l') - \phi(z_1'))\right] + \frac{\alpha}{2} \int_{\eta=0}^{z_L'} \phi(\eta)d\eta$$

$$\geq \frac{\alpha}{2}\left[z_1'\phi(z_1') + \int_{\eta=z_1'}^{z_L'}(\phi(\eta) - \phi(z_1'))d\eta\right] + \frac{\alpha}{2} \int_{\eta=0}^{z_L'} \phi(\eta)d\eta$$

$$\geq \alpha z_1'\phi(z_1') - \frac{\alpha}{2}z_L'\phi(z_1') + \int_{z_1'}^{z_L'} \alpha\phi(\eta)d\eta,$$

where the second and third inequalities follow from Lemma 10, and the fourth inequality holds because ϕ is an increasing function.

Next, we lower-bound $\sum_{j \in B_n \setminus C_n} \Delta_{\Lambda_n}^{(j)}$ as follows:

$$\sum_{j \in B_n \setminus C_n} \Delta_{\Lambda_n}^{(j)} \geq \frac{\alpha}{2}d_n \max\left\{0, y_n - z_L'\right\}.$$

The above inequality holds because, for each request j in $B_n \setminus C_n$, we have $a_j + d_j > a_n + d_n$ by the definitions of B_n and C_n. Then, from Eqs. (37)–(38), for each such j, $\Delta_{\Lambda_n}^{(j)}$ is at least $\frac{\alpha}{3}d_n\hat{x}_j$. Furthermore,

$$\sum_{j \in B_n \setminus C_n} \hat{x}_j \geq \max\left\{0, y_n - z_L'\right\}.$$

We proceed to lower-bound the term u_n. From the update rule in Eq. (36), for each request n we have:

$$u_n \geq \frac{\alpha}{2}(d_n - \phi(y_n + \hat{x}_n/k))\hat{x}_n$$

Combining bounds obtained for the LHS of dual constraint in Eq. (35), we obtain

$$u_n + \sum_{j=n}^{N} \lambda_j \mathbb{I}_{\{a_n+d_n>a_j\}} \geq \frac{\alpha}{2}(d_n - \phi((y_n + \hat{x}_n)/k))\hat{x}_n + \sum_{j \in B_n} \Delta_{\Lambda_n}^{(j)}$$

$$= \frac{\alpha}{2}(d_n - \phi((y_n + \hat{x}_n)/k))\hat{x}_n + \sum_{j \in C_n} \Delta_{\Lambda_n}^{(j)} + \sum_{j \in B_n \setminus C_n} \Delta_{\Lambda_n}^{(j)}$$

$$\geq \frac{\alpha}{2}(d_n - \phi(Z))\hat{x}_n + \alpha z_1'\phi(z_1') - \frac{\alpha}{2}z_L'\phi(z_1') + \int_{z_1'}^{z_L'} \alpha\phi(\eta) \, d\eta + \frac{\alpha}{2}d_n \max\left\{0, Z - z_L'\right\}$$

$$\geq \alpha z_1'\phi(z_1') - \frac{\alpha}{2}z_L'\phi(z_1') + \int_{z_1'}^{z_L'} \alpha\phi(\eta) \, d\eta + \frac{\alpha}{2}d_n \max\left\{0, \phi^*(d_n) - z_L'\right\}$$

$$\geq d_n,$$

where the last inequality follows from the lemma given below. Consequently, we conclude that the set of dual constraints in Eq. (35) is satisfied for each request n given the design of dual constraints in Eqs. (36)–(38). **Therefore, if the increasing pricing function ϕ satisfies the system of constraints in Proposition 7, it establishes the α-competitiveness of Algorithm 5. Thus, we complete the proof of Proposition 7 if the following lemma follows.**

Lemma 11. *If the ϕ function satisfy the constraints in Eqs. (32)–(33) then*

$$\alpha z_1'\phi(z_1') - \frac{\alpha}{2}z_L'\phi(z_1') + \int_{z_1'}^{z_L'}\alpha\phi(\eta)d\eta + \frac{\alpha}{2}d_n\max\left\{0, \phi^*(d_n) - z_L'\right\} \geq d_n,$$

$$\forall d_n \in [d_{\min}, d_{\max}], z_1' \in \left(0, z_L'\right], z_L' \in \left(0, \phi^*(d_{min})\right].$$

Proof. Let us denote the left-hand side of the above inequality by

$$\mathfrak{L}(z_1', z_L', d_n),$$

which is a function of z_1', z_L', and d_n. We are going to prove that

$$\mathfrak{L}(z_1', z_L', d_n) \geq \min\left\{\mathfrak{L}\left(\frac{z_L'}{2}, z_L', d_n\right), \quad \mathfrak{L}(z_L', z_L', d_n)\right\},$$

for all values of $z_1' \in \left(0, z_L'\right]$.

Step 1: Find the critical point in the interior. Fix z_L' and d_n and differentiate L with respect to z_1'. A straightforward calculation shows

$$\frac{\partial\mathfrak{L}}{\partial z_1'} = \frac{\alpha}{2}\phi'(z_1')\left(2z_1' - z_L'\right).$$

Since $\phi'(z_1') \neq 0$, setting this derivative to zero forces

$$z_1' = \frac{z_L'}{2}.$$

A direct substitution $z_1' = \frac{z_L'}{2}$ simplifies the first two terms and yields

$$\mathfrak{L}\left(\frac{z_L'}{2}, z_L', d_n\right) = \int_{z_L'/2}^{z_L'}\alpha\cdot\phi(\eta)d\eta + \frac{\alpha}{2}d_n\max\{0, \phi^*(d_n) - z_L'\}.$$

Step 2: Evaluate $\mathfrak{L}$ on the boundary plane where $z_1' = z_L'$. When $z_1' = z_L'$, the integral term vanishes. One obtains

$$\mathfrak{L}(z_L', z_L', d_n) = \frac{\alpha}{2}z_L'\phi(z_L') + \frac{\alpha}{2}d_n\max\{0, \phi^*(d_n) - z_L'\}.$$

This can be written as

$$\frac{\alpha}{2}d_n\phi^*(d_n) - \frac{\alpha}{2}z_L'\left(d_n - \phi(z_L')\right)$$

for values of $z_L' \leq \phi^*(d_n)$.

Step 3: Evaluate $\mathfrak{L}$ on the boundary plane where z_1' converges to zero. It can be verified that $\mathfrak{L}(z_1', z_L', d_n)$ is lower-bounded by $L(z_L'/2, z_L', d_n)$, as z_1' converges to zero, as follows:

$$\mathfrak{L}(0, z_L', d_n) = \int_0^{z_L'} \alpha \cdot \phi(\eta) d\eta + \frac{\alpha}{2} d_n \max\left\{0, \phi^*(d_n) - z_L'\right\} - \frac{\alpha}{2} z_L' \phi(z_1')$$

$$\geq \int_{z_L'/2}^{z_L'} \alpha \cdot \phi(\eta) d\eta + \frac{\alpha}{2} d_n \max\left\{0, \phi^*(d_n) - z_L'\right\} = L(z_L'/2, z_L', d_n),$$

where the inequality follows from the fact that ϕ function is increasing.

Putting together the above results, in the above three steps, since a continuous function on a closed interval attains its minimum either at a critical point or on the boundary, for each fixed z_L' and d_n we have

$$\mathfrak{L}(z_1', z_L', d_n) \geq \min\left\{\mathfrak{L}\left(\frac{z_L'}{2}, z_L', d_n\right), \quad \mathfrak{L}(z_L', z_L', d_n)\right\}$$

$$= \min\left\{\frac{\alpha}{2} d_n \phi^*(d_n) - \frac{\alpha}{2} z_L'\left(d_n - \phi(z_L')\right),\right.$$

$$\left. \int_{z_L'/2}^{z_L'} \alpha \cdot \phi(\eta) d\eta + \frac{\alpha}{2} d_n \max\{0, \phi^*(d_n) - z_L'\}\right\}$$

$$\geq d_n,$$

where the last inequality follows from the constraints in Eq. (32) and Eq. (33). This completes the proof of the lemma. □

J Connection to the Results of [6]

In the following, building on the techniques from [6], we reproduce their result for the fractional version of the kRental-Variable problem using our pseudo-utility maximization approach. We focus on a special case in which rental durations are restricted to integer values and requests arrive at discrete time steps. Accordingly, for each request n, we define its rental period as $\mathcal{T}_n := \{a_n, a_n+1, \ldots, a_n+d_n-1\}$. Furthermore, we make the following assumption where:

Assumption 3. *The rental duration of each request n, d_n is integral and bounded, i.e., $d_n \in \{1, 2, \ldots, d_{\max}\}, \forall n \in [N]$.*

In their work, they consider a parametrized pricing function in the following form

$$\phi(y) = \eta(\beta^{\frac{y}{k}} - 1), y \in [0, k], \tag{42}$$

where $\eta > 0$ and $\beta \geq e$ are two parameters to be determined.

We propose Algorithm 6 below, which is also referred to as online forward looking price-based algorithm, FLP-ϕ-VARIABLE.

Algorithm 6: FLP-ϕ-VARIABLE for Fractional kRental-Variable

---

1 **Input:** Pricing function $\phi(\cdot)$

2 **Initialization:** Set utilization $y_t^{(0)} = 0$ for all t.

3 **while** *item n arrives* **do**

4   Observe item n's request (a_n, d_n).

5   Determine $\hat{x}_n$ by solving the pseudo-utility maximization problem:

$$\hat{x}_n = \arg\max_{x \in [0,1]} \left\{ x \cdot d_n - \sum_{t \in \mathcal{T}_n} \int_{y_t^{(n-1)}}^{y_t^{(n-1)}+x} \phi(u)\, du \right\}. \tag{43}$$

6   Update the utilization profile:

$$y_t^{(n)} = \begin{cases} y_t^{(n-1)} + \hat{x}_n & \text{if } t \in \mathcal{T}_n, \\ y_t^{(n-1)} & \text{otherwise.} \end{cases}$$

7 **end**

Similar to Algorithm 4, FLP-ϕ-VARIABLE employs a pricing function ϕ to determine the fractional allocation $\hat{x}_n$ for each arriving request n. However, it is important to highlight two key differences between Algorithm 6 and Algorithm 4.

- **Duration-dependent *vs* Duration-oblivious**. It is evident that the pseudo-utility maximization problem in Eq. (43) accounts for the item's utilization over the entire duration $\mathcal{T}_n$ of request n. Consequently, the pseudo-cost is explicitly *duration-dependent*. This is in sharp contrast to the *duration-oblivious* nature of Algorithm 4, introduced in Section 4 for the kRental-Variable problem, which considers only the item's utilization at the arrival time of each request n.

- **Different design principles to ensure feasibility**. In FLP-ϕ-VARIABLE, the online decision is solely determined by the pseudo-utility maximization problem, with no additional mechanism to explicitly enforce capacity constraints. This represents another key difference between the design principles of Algorithm 6 and Algorithm 4. Intuitively, if the pricing function is designed to be sufficiently steep (i.e., with large enough η and β), the feasibility of the online solution can be guaranteed. However, increasing η and β also leads to a larger competitive ratio, which is undesirable. Lemma 12 provides a tighter condition on η and β to ensure the feasibility of FLP-ϕ-VARIABLE.

Lemma 12. *FLP-ϕ-VARIABLE is $(1+\eta)\ln(\beta)$-competitive and generates feasible online solutions, with respect to the resource constraint, if the parameters of the pricing function satisfies*

$$\ln \beta \geq -\ln\left(\prod_{i \in [d_{\max}]} \left(1 - \frac{1}{i(1+\eta)}\right) \right). \tag{44}$$

Proof. In the following, we first prove that FLP-ϕ-VARIABLE generates a feasible solution and then we prove the competitive ratio of the algorithm.

Feasibility of FLP-ϕ-VARIABLE. We only need to show that after processing each request n, the utilization at the arrival time a_n is no larger than k, i.e., $y_{a_n}^{(n)} \leq k, \forall n \in [N]$ since $y_{a_n}^{(n)}$ is the largest utilization over time after processing each request n in the online k-rental problem. To show this, we prove a stronger

claim that after processing request n, the utilization difference between two time slots $a_n + \tau$ and $a_n + d$ is upper bounded as follows

$$y_{a_n+\tau}^{(n)} - y_{a_n+d}^{(n)} \leq -\frac{k}{\ln \beta} \ln \left(\prod_{i \in [d-\tau]} (1 - \frac{1}{i(1+\eta)}) \right), \quad 0 \leq \tau \leq d. \tag{45}$$

Since the rental period of an item is at most $d_{\max}$, we have $y_{a_n+d_{\max}}^{(n)} = 0$. If the above claim in Eq. (45) and the condition in Eq. (44) both hold, we can show the online solution is feasible by noting

$$y_{a_n}^{(n)} = y_{a_n}^{(n)} - y_{a_n+d_{\max}}^{(n)} \leq -\frac{k}{\ln \beta} \ln \left(\prod_{i \in [d_{\max}]} (1 - \frac{1}{i(1+\eta)}) \right) \leq k,$$

where the first and second inequalities are due to Eqs. (45) and (44), respectively.

Next, we prove the claim in Eq. (45) by induction. We refer to the processing of request n as step n. In the base case when $n = 0$, the initial utilization of all slots is $y_t^{(0)} = 0, \forall t \in [T]$, and the claim holds. Now suppose the claim holds in step $n - 1$. Consider the following cases.

Case I. Request n is completely rejected, i.e., $\hat{x}_n = 0$. In this case, $y_t^{(n-1)} = y_t^{(n)}, \forall t \in [T]$, and thus the claim holds in step n.

Case II. Request n is partially or completely accepted, i.e., $\hat{x}_n \in (0, 1]$. Based on the pseudo-utility maximization in FLP-ϕ-VARIABLE, we must have $v_n \geq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$ (if $\hat{x}_n < 1$, $v_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$; if $\hat{x}_n = 1$, $v_n \geq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$). Then we have

$$d_n \geq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) = \sum_{t \in \mathcal{T}_n} \eta(\beta^{\frac{y_t^{(n)}}{k}} - 1). \tag{46}$$

Note that $y_{a_n+\tau}^{(n)} - y_{a_n+d}^{(n)} = (y_{a_n+\tau}^{(n-1)} + \hat{x}_n) - (y_{a_n+d}^{(n-1)} + \hat{x}_n) = y_{a_n+\tau}^{(n-1)} - y_{a_n+d}^{(n-1)}$ if $a_n + \tau, a_n + d \in \mathcal{T}_n = \{a_n, \ldots, a_n + d_n - 1\}$ and $y_{a_n+\tau}^{(n)} - y_{a_n+d}^{(n)} = y_{a_n+\tau}^{(n-1)} - y_{a_n+d}^{(n-1)}$ if $a_n + \tau, a_n + d \in \{a_n + d_n, \ldots, T\}$. In these cases, the utilization difference between $y_{a_n+\tau}^{(n)}$ and $y_{a_n+d}^{(n)}$ is the same as the previous step, and the induction holds. Thus, the only interesting case is when

$$a_n + \tau < a_n + d_n \leq a_n + d,$$

where the utilization of slot $a_n + \tau$ increases due to newly leased item while the utilization of $a_n + d$ remains unchanged.

From Eq. (46), we can have

$$\frac{1}{\eta} d_n \geq \sum_{t=a_n}^{a_n+d_n-1} (\beta^{\frac{y_t^{(n)}}{k}} - 1) \tag{47a}$$

$$= \sum_{t=a_n}^{a_n+\tau-1} (\beta^{\frac{y_t^{(n)}}{k}} - 1) + \sum_{t=a_n+\tau}^{a_n+d_n-1} (\beta^{\frac{y_t^{(n)}}{k}} - 1) \tag{47b}$$

$$\geq -d_n + \tau \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} + \sum_{t=a_n+\tau}^{a_n+d_n-1} \beta^{\frac{y_t^{(n)}}{k}} \tag{47c}$$

$$\geq -d_n + \tau \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} + \sum_{t=a_n+\tau}^{a_n+d_n-1} \beta^{\frac{y_{a_n+\tau}^{(n)}}{k} + \frac{1}{\ln \beta} \ln\left(\prod_{i \in [t-a_n-\tau]} (1 - \frac{1}{i(1+\eta)})\right)} \tag{47d}$$

$$= -d_n + \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} \left[\tau + \sum_{t'=0}^{d_n-\tau-1} \prod_{i \in [t']} (1 - \frac{1}{i(1+\eta)}) \right] \tag{47e}$$

$$= -d_n + \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} \left[\tau + \frac{d_n - \tau}{1 - \frac{1}{1+\eta}} \cdot \prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}) \right] \tag{47f}$$

$$= -d_n + \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} \cdot \frac{1+\eta}{\eta} \left[\tau \cdot \frac{\eta}{1+\eta} + (d_n - \tau) \cdot \prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}) \right] \tag{47g}$$

$$\geq -d_n + \beta^{\frac{y_{a_n+\tau}^{(n)}}{k}} \cdot \frac{1+\eta}{\eta} \cdot d_n \cdot \prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}), \tag{47h}$$

which gives

$$y_{a_n+\tau}^{(n)} \leq -\frac{k}{\ln \beta} \cdot \ln \prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}).$$

In the above set of equations, the inequality (47c) holds since the utilization is monotonically non-increasing over time, and thus $y_t^{(n)} \geq y_{a_n+\tau}^{(n)}, \forall t = a_n, \ldots, a_n + \tau - 1$. The inequality (47d) is obtained by applying the induction hypothesis, i.e., $\forall t = a_n + \tau, \ldots, a_n + d_n - 1$,

$$y_t^{(n)} = \hat{x}_n + y_t^{(n-1)} \geq \hat{x}_n + y_{a_n+\tau}^{(n-1)} + \frac{k}{\ln \beta} \ln \left(\prod_{i \in [t-a_n-\tau]} (1 - \frac{1}{i(1+\eta)}) \right).$$

The equality (47f) holds due to the following equation that can be shown by induction:

$$\sum_{t=0}^{b-1} \prod_{i \in [t]} (1 - \frac{a}{i}) = \frac{b}{1-a} \prod_{i \in [b]} (1 - \frac{a}{i}), \forall b \in \mathbb{N}, a \geq 0.$$

The last inequality holds since $\prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}) \leq \frac{\eta}{1+\eta}$, where $d_n - \tau \geq 1$.

Thus, we have

$$y_{a_n+\tau}^{(n)} - y_{a_n+d}^{(n)} \leq y_{a_n+\tau}^{(n)} \leq -\frac{k}{\ln \beta} \cdot \ln \prod_{i \in [d_n-\tau]} (1 - \frac{1}{i(1+\eta)}) \leq -\frac{k}{\ln \beta} \cdot \ln \prod_{i \in [d-\tau]} (1 - \frac{1}{i(1+\eta)}), \tag{48}$$

which shows that the induction holds in step n. This completes the feasibility proof.

Proof of $(1+\eta)\cdot\ln(\beta)$-competitiveness of FLP-ϕ-VARIABLE. We consider the following relaxed primal problem and its corresponding duel problem:

$$\max_{x_n \geq 0} \quad \sum_{n \in [N]} d_n x_n \tag{49a}$$

$$\text{s.t.} \quad \sum_{n \in [N]:t \in \mathcal{T}_n} x_n \leq k, \forall t \in [T], \quad (\lambda_t) \tag{49b}$$

$$x_n \leq 1, \forall n \in [N], \quad (\gamma_n) \tag{49c}$$

$$\min_{\lambda_t, \gamma_n \geq 0} \quad k \sum_{t \in [T]} \lambda_t + \sum_{n \in [N]} \gamma_n \tag{50a}$$

$$\text{s.t.} \quad d_n \leq \gamma_n + \sum_{t \in \mathcal{T}_n} \lambda_t, \forall t \in [T], n \in [N]. \tag{50b}$$

We construct a set of feasible dual variables as follows:

$$\lambda_t = \phi(y_t^{(N)}), \qquad \forall t \in [T],$$

$$\gamma_n = \begin{cases} 0 & 0 \leq \hat{x}_n < 1, \\ d_n - \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) & \hat{x}_n = 1, \end{cases} \qquad \forall n \in [N],$$

where $\hat{x}_n$ is the online solution of FLP-ϕ-VARIABLE and $y_t^{(n)}$ is the utilization of time t after processing the n-th item by FLP-ϕ-VARIABLE. We show in the following that the constructed dual solution is feasible:

Case I. If $x = 0$, then we have $\gamma_n = 0$ and $d_n \leq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$ (based on (43)). Thus,

$$d_n \leq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) \leq \sum_{t \in \mathcal{T}_n} \phi(y_t^{(N)}) = \sum_{t \in \mathcal{T}_n} \lambda_t + \gamma_n.$$

Case II. If $0 < \hat{x}_n < 1$, then we have $\gamma_n = 0$ and $d_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$. Thus,

$$d_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) \leq \sum_{t \in \mathcal{T}_n} \lambda_t + \gamma_n.$$

Case III. If $\hat{x}_n = 1$, then we have $\gamma_n = d_n - \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) \geq 0$. Thus,

$$d_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) + \gamma_n \leq \sum_{t \in \mathcal{T}_n} \lambda_t + \gamma_n.$$

The dual objective at the feasible dual solution provides an upper bound on the relaxed primal and the original primal objective. Thus, we have

$$\text{OPT} \leq k \sum_{t \in [T]} \phi(y_t^{(N)}) + \sum_{n \in [N]} \gamma_n$$

$$= \sum_{n \in [N]} \left[k \sum_{t \in \mathcal{T}_n} (\phi(y_t^{(n)}) - \phi(y_t^{(n-1)})) + \gamma_n \right],$$

which holds as $\phi(0) = 0$. We next show that $k \sum_{t \in \mathcal{T}_n} (\phi(y_t^{(n)}) - \phi(y_t^{(n-1)})) + \gamma_n \leq (1+\eta) \ln \beta \cdot \hat{x}_n d_n, \forall n \in [N]$.

Consider the following three cases.

Case I. If $\hat{x}_n = 0$, then we have $k \sum_{t \in \mathcal{T}_n} (\phi(y_t^{(n)}) - \phi(y_t^{(n-1)})) + \gamma_n = (1 + \eta) \ln \beta \cdot \hat{x}_n d_n = 0$.

Case II. If $0 < \hat{x}_n < 1$, we have $\gamma_n = 0$ and $d_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$. This gives

$$
\begin{aligned}
k \sum_{t \in \mathcal{T}_n} (\phi(y_t^{(n)}) - \phi(y_t^{(n-1)})) &= k\eta \sum_{t \in \mathcal{T}_n} \beta^{\frac{y_t^{(n)}}{k}} (1 - \beta^{-\frac{\hat{x}_n}{k}}) \\
&\leq k\eta \sum_{t \in \mathcal{T}_n} \beta^{\frac{y_t^{(n)}}{k}} \frac{\hat{x}_n}{k} \ln \beta \\
&= \ln \beta \cdot \sum_{t \in \mathcal{T}_n} \hat{x}_n \eta [\beta^{\frac{y_t^{(n)}}{k}} - 1] + \ln \beta \cdot \eta \sum_{t \in \mathcal{T}_n} \hat{x}_n \\
&= \ln \beta \cdot \hat{x}_n \cdot \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) + \ln \beta \cdot \eta \cdot \hat{x}_n \cdot d_n \\
&\leq \ln \beta \cdot \hat{x}_n d_n + \ln \beta \cdot \eta \cdot \hat{x}_n d_n = (1 + \eta) \ln \beta \cdot \hat{x}_n d_n,
\end{aligned}
$$

where the last inequity holds since $d_n = \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$.

Case III. If $\hat{x}_n = 1$, we have $\gamma_n = d_n - \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$ and $d_n > \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)})$. This gives

$$
\begin{aligned}
k \sum_{t \in \mathcal{T}_n} (\phi(y_t^{(n)}) - \phi(y_t^{(n-1)})) + \gamma_n &= k\eta \sum_{t \in \mathcal{T}_n} \beta^{\frac{y_t^{(n)}}{k}} (1 - \beta^{-\frac{1}{k}}) + \gamma_n \\
&\leq k\eta \sum_{t \in \mathcal{T}_n} \beta^{\frac{y_t^{(n)}}{k}} \frac{1}{k} \ln \beta + \gamma_n \\
&= \ln \beta \cdot \sum_{t \in \mathcal{T}_n} \eta [\beta^{\frac{y_t^{(n)}}{k}} - 1] + \ln \beta \cdot \eta \cdot d_n + \gamma_n \\
&= \ln \beta \cdot \sum_{t \in \mathcal{T}_n} \phi(y_t^{(n)}) + \ln \beta \cdot \eta \cdot d_n + \gamma_n \\
&= \ln \beta \cdot (d_n - \gamma_n) + \ln \beta \cdot \eta \cdot d_n + \gamma_n \\
&\leq \ln \beta \cdot d_n + \ln \beta \cdot \eta \cdot d_n + (1 - \ln \beta)\gamma_n \\
&\leq (1 + \eta) \ln \beta \cdot d_n,
\end{aligned}
$$

where the last inequality holds when $\beta \geq e$.

This completes the proof of Lemma 12. □

Based on Lemma 12, we can design the parameters of the pricing function by formulating and solving the following optimization problem, which aims to minimize the competitive ratio of FLP-ϕ-VARIABLE:

$$
\min_{\beta \geq e, \eta > 0} \quad (1 + \eta) \ln \beta \tag{51a}
$$

$$
\text{s.t.} \quad \ln \beta \geq - \ln \left(\prod_{i \in [d_{\max}]} (1 - \frac{1}{i(1 + \eta)}) \right). \tag{51b}
$$

Thus, using the standard online primal-dual analysis, the results of [6] can be extended to the *fractional* setting through the same pseudo-utility maximization framework as in Algorithm 3 and Algorithm 4.

We end this section with a discussion of some interesting future directions. Based on the results of this work and those of [6], a natural direction for future work is to develop a rounding scheme that converts the fractional algorithm such as Algorithm 6 into integral decisions, thereby addressing the integral version of the kRental-Variable problem. However, due to the impossibility result in Theorem 3, any such rounding scheme will necessarily incur some loss in performance. An alternative and promising direction is to adopt the relax-and-round approach from Section 4, where the rounding step is integrated into the design of the fractional solution. This would require a more sophisticated treatment of the pricing function ϕ, both to ensure feasibility and to optimize the competitive ratio.