Randomized Posted Pricing and Rounding Schemes for Online Selection and Matching

by

Hossein Nekouyan Jazi

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Motivated by real-world online markets such as online advertising and cloud computing, this thesis investigates adversarial online resource allocation problems. Specifically, we focus on settings in which a decision-maker controls a limited supply of resources and must make immediate, irrevocable allocation decisions upon the sequential arrival of buyers, each possessing distinct valuations. Uncertainty regarding future buyer valuations and overall market demand introduces inherent complexity into these online decision-making scenarios.

To address these challenges, we study two primary adversarial online allocation models: *online selection* and *online matching*. Both models consider a scenario in which the decision-maker, constrained by limited units of resources, encounters buyers sequentially—each with individual valuations—and must allocate resources immediately upon each buyer's arrival. Our primary focus is on a class of online selection and matching problems characterized by *diseconomy-of-scale* (i.e., increasing difficulty or cost associated with producing additional units of resources) and resource *reusability* (i.e., allocated resources become available again after a certain period). For each of these problems, this thesis develops randomized algorithms achieving optimal or near-optimal competitive guarantees. From a technical perspective, our approach builds upon two core methodologies. First, we show that the challenge posed by diseconomy-of-scale can be effectively managed through randomized dynamic *posted-price mechanisms*, wherein the seller sets resource-specific prices, allowing buyers to select options maximizing their individual utilities in a take-it-or-leave-it fashion. Second, we adopt the *relax-and-round* framework, initially employing resource-specific pricing functions to generate fractional allocations, which are subsequently converted into integral decisions via novel

online rounding procedures. This combined approach effectively addresses uncertainty related to future buyer valuations and resource demands, contributing deeper insights into optimal resource allocation strategies in dynamic online markets.

Preface

Parts of this thesis are based on two previously published papers. Chapter 2 is based on the paper published in the 2025 ACM Web Conference (WWW 2025) [JSBT25]. Chapter 4 is based on the paper published in the 2024 Conference on Web and Internet Economics (WINE 2024) [SJTB24]. Chapters 3, 4, and 6 are based on ongoing work that is part of a larger project and is intended for future publications.

Acknowledgements

I would like to thank my supervisor, Dr. Xiaoqi Tan, for his support, guidance, and helpful advice during my research. I really appreciate his patience and the time he spent helping me improve my work. It has been a great experience working with him. Under his mentorship, I took my first steps as a researcher and learned not only how to conduct research, but also how to write, present, and communicate my work effectively.

I would also like to thank Dr. Bo Sun for his collaboration and support. His input and feedback were very valuable. I learned a lot from his approach to research and problem-solving.

I would also like to thank my family for their constant love, support, and encouragement throughout this journey. I am especially grateful to my dear friend Mojtaba for his continuous support and motivation to help me grow and improve as a researcher. At the end, I would like to dedicate this work to the memory of my late grandfather and grandmother, whom I lost recently.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Online resource allocation—the process of assigning limited resources to sequential online requests to maximize social welfare or profit—has been a central research topic in computer science and operations research. This area has significant applications in domains such as cloud computing [ZLW17, ZHW$^+$15], network routing [CST22, AAP93, BN09], and various market-driven online platforms [OP12, KFV15]. A common assumption in existing literature is that a seller or decision-maker possesses a finite inventory of resources that must be allocated to a sequence of online buyers, each with distinct valuations. Typical objectives in these problems include maximizing social welfare (i.e., the aggregate valuations of buyers who receive resources) or optimizing revenue generation for the seller. The inherent challenge arises from the irrevocable nature of online decisions, necessitating algorithmic strategies that effectively balance efficiency and competitiveness under uncertainty.

Recent studies have introduced more generalized mathematical models that more precisely capture the dynamics of online markets, broadening their applicability to diverse real-world scenarios. For example, researchers have examined situations where decision-makers face *diseconomies of scale* in resource provisioning—that is, scenarios in which the marginal cost of supplying each additional unit increases. In cloud computing systems, for instance, server power costs grow superlinearly with increased resource utilization [AAZ16]. Similarly, congestion costs such as end-to-end delay in network routing escalate significantly with higher traffic intensity. In Chapter 2, we study the online selection problem under diseconomies of scale, which models an online market in which sellers face increasing marginal production costs. We develop algorithms that effectively address these challenges while achieving strong performance guarantees.

Furthermore, inspired by real-world applications such as Internet advertising, personalized recommendations, and crowdsourcing, recent research has proposed theoretical models—such as the online assortment planning problem and online matching (and their various variants)—that capture the inherent trade-offs and complexities of these markets. In these scenarios, a seller must dynamically select an optimal subset of products or services from a finite, multi-resource inventory to present to incoming buyers. The primary objective is to maximize revenue or social welfare by aligning the offered assortment with each buyer's valuations while simultaneously maintaining balanced inventory levels across resource types. Consequently, substantial research has focused on developing dynamic assortment and pricing strategies that deliver strong performance outcomes while preserving inventory balance. In Chapters 5 and 6, we investigate problems within this context.

Unlike the aforementioned studies that primarily address permanent resource allocation, recent research has shifted attention to models where resources can be *reused* multiple times. For instance, companies such as Amazon, Google, and Microsoft Azure offer cloud computing services in which users rent computing capacity for a specified period before returning it. Similarly, firms like Rent the Runway and Glam Corner rent fashion items, and companies such as CubeSmart and MakeSpace lease storage units that customers return after use. Resource reusability introduces additional complexity to real-time resource allocation, as platforms must accommodate overlapping usage periods and fluctuating demand while maintaining high service quality and profitability. Motivated by these applications, we study online markets capturing the dynamics of allocating reusable resources in Chapters 3, 4, and 6.

To design online algorithms with performance guarantees, researchers have examined online allocation problems under different *arrival models*. For example, the secretary problem [Gar70] assumes that prices arrive in a uniformly random order, while the prophet inequality [SC84] assumes that prices are drawn from known distributions. In contrast to these stochastic models, our work in this thesis focuses on an *adversarial setting* where buyer valuations can be arbitrary within a bounded range, and we do not assume any prior knowledge about the demand for different resources over an unknown horizon. This setting has been widely studied in classic online search problems [EYFKT01, LPS09, JLTZ21a] and the online knapsack problem [ZCL08, SYH+22], and it has various applications in revenue management [MSL20, BQ09]. Furthermore, in this thesis, we evaluate the performance of online algorithms using the *competitive ratio* as our primary metric [BEY98]. The competitive ratio measures the worst-case performance of an online algorithm relative to an optimal offline solution that has complete information. Specifically, an online algorithm $\texttt{ALG}$ is said to be α-competitive if, for any input instance I of the problem, the following inequality holds:

$$\alpha \geq \frac{\texttt{OPT}(I)}{\mathbb{E}[\texttt{ALG}(I)]}, \tag{1.1}$$

where the expectation $\mathbb{E}[\texttt{ALG}(I)]$ is taken over the randomness of the online algorithm, and $\texttt{OPT}(I)$ denotes the performance of the optimal offline clairvoyant algorithm that has the full knowledge of instance I.

1.1 Problem Overview

In this section, we provide an overview of the problems studied in Chapters 2–6 and illustrate their connections to the broader literature on online optimization under adversarial arrival models.

1.1.1 `kSelection-cost`: Online Selection with Diseconomy of Scale

In this problem, we study online allocation with increasing marginal production costs. In particular, we frame it as an online k-selection with diseconomies-of-scale (`kSelection-cost`) in a posted price mechanism: A seller offers a certain resource to buyers arriving one at a time in an online manner. Each buyer has a private valuation v_n for one unit of the resource. The seller can produce k units of the resource in total; however, the marginal cost of producing each unit increases as more units are produced. When the n-th buyer arrives, the seller posts a price p_n to the buyer, provided that fewer than k units have already been produced and allocated. If the buyer's valuation v_n exceeds p_n, the buyer accepts the price and takes one unit of the resource. The objective is to maximize social welfare, defined as the sum of the utilities of all the buyers and the revenue of the seller.

A key challenge in these problems is balancing pricing strategies. Setting prices too low early on may allocate many resources to low-value buyers, increasing production costs and lowering social welfare. Conversely, overly high prices can result in missed opportunities to sell. Thus, pricing for k units must carefully account for early-stage decisions to avoid rapid growth in marginal production costs while maximizing efficiency.

Despite previous efforts [HK19, TYBLG23, SJTB24], two questions remain unresolved: First, how to derive a tight lower bound for `kSelection-cost` in small inventory settings (i.e., when k is finite and small)? Second, it remains an open question how to develop randomized algorithms to solve `kSelection-cost` with tight guarantees, especially for settings when k is small. In Chapter 2, we address these questions by deriving a new tight lower bound for the `kSelection-cost` problem, achieving the best-known results in both small and asymptotically large inventory settings.

1.1.2 `kRental-fixed` & `kRental-variable`: Online k-Rental with Fixed and Variable Rental Durations

In Chapters 3 and 4, we study two variants of adversarial online allocation problems with reusable resources. In both problems, a seller (or decision maker) maintains an inventory of k identical units of a resource. A sequence of buyers arrives at arbitrary times throughout the horizon, and each buyer requests to *rent* one unit of the resource for a specified duration. If a unit is allocated to a buyer, it becomes available for reallocation only after that buyer's rental period ends. Additionally, each buyer has a unique valuation for renting a unit of the resource. Our objective is to maximize the total valuation of all accepted buyer requests.

- In Chapter 3, we examine the `kRental-fixed` problem, wherein every buyer's rental duration is fixed and identical across all buyers, but there is uncertainty concerning each buyer's valuation for renting a unit of the resource over that fixed duration.
- In Chapter 4, we focus on `kRental-variable`, where buyers' rental durations vary and are buyer-dependent. Their valuations are assumed to be linear in their rental durations.

In these two problems, the uncertainty over buyers' valuations and the inherent trade-off between accepting early arrivals with lower valuations versus waiting for potentially higher-valued buyers makes the decision process challenging. Furthermore, unlike in non-reusable online resource allocation settings such as the `kSelection-cost` problem studied in Chapter 2, the reusability of the resource adds extra complexity. An algorithm must schedule the allocation of resource units efficiently to maximize their usage time, yet it must do so without precise knowledge of how future requests may overlap, further complicating the scheduling process.

Recently, [DFNS22] studied online bipartite matching with reusability in a setting where the rental duration of each buyer is fixed in all types of resources. Their first algorithm achieves a competitive ratio of 0.589, without any discussion of optimality or lower bounds for online algorithms. They also designed a second algorithm that uses a correlation-based rounding scheme, which obtains a 0.505-competitive ratio—close to the 0.5 ratio of the greedy algorithm in that setting. The main limitation of deriving such an algorithm that performs very close to greedy is that the rounding scheme is not sufficiently effective at rounding the fractional solution. In Chapter 3, we introduce an novel problem which examines a special case of the problem studied in [DFNS22], in which the decision maker holds only one resource type, yet uncertainty persists regarding each buyer's valuation. We investigate whether a tight competitive ratio can be achieved under these assumptions. Furthermore, we aim to determine whether more powerful correlation schemes can be devised to overcome the limitations observed in [DFNS22] to round fractional solutions in this setting.

Furthermore, [HC23, GG20] studied adversarial online allocation with reusable resources in which the rental durations vary and are buyer-dependent. Their algorithms attain order-optimal results in large-inventory settings, where each resource type's inventory asymptotically converges to infinity. Whether one can design algorithms with strong performance guarantees for the small-inventory setting remains an open problem. In addition, [GG20] proposed a static threshold algorithm that achieves near-optimal performance guarantees for the `kRental-variable` problem. However, a static threshold approach may not be well-suited for online resource allocation with reusable resources, as the algorithm must address more complex underlying dynamics and perform scheduling to maximize the utilization time of different resources. In Chapter 4, we investigate whether a dynamic pricing scheme—one that continuously adjusts resource prices to adapt to evolving market conditions—can yield stronger performance guarantees in this setting. Furthermore, we design an online rounding scheme that, when combined with the dynamic pricing approach, efficiently rounds fractional solutions to achieve robust performance guarantees.

1.1.3 `Matching-BW`: Online Edge-Weighted Matching with Bounded Weights

In Chapter 5, we study adversarial online edge-weighted matching with bounded weights through posted pricing mechanisms: the seller discloses a unit price for each resource type before each buyer's arrival. Acting as price takers, buyers decide to purchase one resource if the utility (i.e., the buyer's valuation of the resource minus the posted price) is non-negative. Our goal is to again maximize the social welfare (i.e., the sum of the seller's revenue and the buyers' utility) in the posted pricing mechanism, which also aligns with maximizing the valuations of all selected buyers. Compared to the previous chapters, where we study different variants of online selection problems, in this setting the seller manages multiple resource types. Furthermore, the seller can allocate (match) at most one unit of a resource type to each buyer. Thus, a new layer of uncertainty arises in this problem with respect to the demand for different resource types, and the algorithm must balance the inventory available from the various resource types to address this uncertainty.

There has been a stream of literature that studies how to design posted prices to attain (near) optimal guarantees for adversarial online selection problems (e.g., [ZLW17, TSLG+20, MSL20]). Nearly all of these works rely on dynamic pricing strategies, which publish different prices for different buyers to make the best use of pricing power. However, dynamic pricing raises additional concerns in practice since it introduces *price discrimination*, i.e., the same resource is sold at different prices for different buyers who arrive at different times. This incentivizes buyers to make strategic purchase decisions. For example, in the airline ticket and hotel room booking problems, where dynamic pricing has been widely applied, each buyer may query the same flight ticket or hotel room multiple times, seeking a lower posted price. Price discrimination may be advantageous for the seller to maximize revenue in the short term. However, it is inherently unfair to buyers and diminishes the buyer experience, potentially leading to adverse effects in the long run.

Algorithms with static pricing are not price-discriminatory because they maintain fixed prices for each resource type throughout the entire buyer arrival horizon. However, to handle market uncertainty, they employ randomization when setting prices. In Chapter 5, we study static pricing for the `Matching-BW` problem. Our key questions are: How can we design the best possible static pricing strategy, and can static pricing achieve performance comparable to dynamic pricing in adversarial settings?

1.1.4 `Matching-RR`: Online Edge-Weighted Matching with Reusable Resources

In Chapter 6, we extend the `kRental-variable` problem (originally introduced in Chapter 4) to an online edge-weighted matching setting. Instead of having a single resource type, the decision maker now has multiple resource types, each with a supply of identical units. Upon arrival, a buyer reports a vector indicating the rental durations for each resource type of interest, along with a per-unit-time valuation for each type. The buyer's valuation for obtaining a unit of a resource type is given by the product of the requested duration and the corresponding per-unit-time valuation for that resource type. The decision maker can allocate (i.e., match) at most one unit of one resource type to each buyer, and the objective is to maximize the total value obtained. Compared to `kRental-variable` in Chapter 4, the decision maker faces additional uncertainty with respect to both the buyers' per-unit-time valuations and the need to balance allocations among different resource types.

[SYH+23] studied a related problem within the framework of the online knapsack problem with departures, proposing a threshold-based algorithm that attains an order-optimal result. Their analysis, however, relies on large-inventory assumptions and restricts buyers to arriving in discrete time steps. Similar works, such as [HC23, FNS19], also consider large-inventory settings and discrete arrival models. In contrast, our focus is on designing algorithms whose performance guarantees are independent of the inventory size, thereby maintaining robustness across both small- and large-inventory regimes. Moreover, in the setting considered in Chapter 6, buyers may arrive at any point of time over a continuous horizon. We also investigate the development of effective relax-and-round algorithms, which provide strong performance guarantees through online rounding schemes. Finally, we explore how to integrate the ideas introduced in Chapters 4 and 6, given that `Matching-RR` must manage uncertainties stemming from buyer-specific rental requests (including per-unit-time valuations) as well as multiple resource types.

1.2 Related Literature

Online resource allocation has been extensively studied in recent years under various arrival models and assumptions. For example, the problem has been formulated as the secretary problem under the random-order model [Gar70, GS21, HS24] and as the prophet inequality problem under a stochastic input model [SC84, Luc17, CFH+19]. Moreover, beyond the classical i.i.d. stochastic setting of the prophet inequality, recent work has explored models with correlated arrivals based on Markov chains [JLL+23]. In contrast to these assumptions based on statistical models, our focus lies on the adversarial setting, where arrivals' valuations can be arbitrary within a bounded support. We will now delve into reviewing the related work concerning adversarial online allocation problems.

Online knapsack problem with bounded values. The adversarial online knapsack problem has been extensively studied in the large-supply regime, where the number of identical resource units approaches infinity. [ZCL08] examined this setting with infinitesimal item weights and bounded value density, proposing a threshold-based algorithm that achieves the optimal competitive ratio among all online algorithms. Subsequent works have shown that this threshold-based approach can be interpreted as a deterministic dynamic pricing algorithm [ZLW17], and have extended it to more complex settings, including multiple and multi-dimensional knapsack problems [SYH+22]. These studies primarily focus on dynamic pricing under the assumption of infinitesimally small item sizes.

In the small-inventory regime, [JLTZ21a] presented an optimal randomized static pricing algorithm for the single-unit (unit-supply) case. Further extending this line of work, [SJTB24] generalized the static pricing approach to accommodate larger inventory sizes. In Chapters 2, 3, and 4, we investigate different variants of online selection problems that incorporate diseconomies of scale, reusability with fixed duration, and, in the most general setting, buyer-dependent durations.

Online selection with diseconomy-of-scale (`kSelection-cost`). `kSelection-cost` is another extension of the online selection problem that accounts for a non-linear (convex) supply cost. The incorporation of increasing marginal production costs into online resource allocation is first introduced by [BGMS11]. Subsequently, [HK19] studied a similar problem, termed online combinatorial auction with convex cost, focusing specifically on combinatorial auctions in the large-supply regime. Variants of `kSelection-cost` have since been explored, including online convex packing and covering [ABC+16] and [TSLG+20].

To address the challenges in `kSelection-cost`, [HK19] developed optimal *deterministic dynamic pricing* mechanisms for fractional online combinatorial auctions with production costs and infinite capacity ($k = \infty$). They extended this to the integral case using fractional pricing functions, achieving a competitive ratio close to the fractional setting but with a nonzero additive loss. However, as the competitive ratio approaches the fractional lower bound, the additive loss grows unbounded, which is undesirable. To overcome this, Tan et al. [TYBLG23] studied online selection with convex costs and limited supply ($k < \infty$ where k is number of units of the resource), establishing a lower bound for the integral setting without additive loss. They further showed that the competitive ratio of their deterministic posted price mechanism asymptotically converges to the lower bound as k grows large. Recently, Sun et al. [SJTB24] proposed a *randomized static pricing* algorithm, which samples a static price from a pre-determined distribution for `kSelection-cost`. This randomization improves performance over the deterministic approach in small inventory settings but is not asymptotically optimal and fails to converge to the lower bound from [TYBLG23] as $k \to \infty$.

Recent years have seen efforts to study online allocation problems with various forms of production costs in stochastic settings (e.g., [BMY15], [GMM18], [BUCM12], [Sek17]). For instance, [BMY15] examined online allocation with economies of scale (decreasing marginal costs), proposing a constant-competitive strategy for unit-demand customers with valuations sampled i.i.d. from an unknown distribution. In contrast, [Sek17] addressed bayesian online allocation with convex production costs, developing posted price mechanisms with $O(1)$-approximation for fractionally subadditive buyers and logarithmic approximations for subadditive buyers. Our study differs by focusing on `kSelection-cost` in adversarial settings, assuming no knowledge of the arrival sequence beyond the finite support of valuations, making these results not directly comparable to ours.

Online interval scheduling. The adversarial online interval scheduling problem, introduced by [LT94], considers scheduling a sequence of intervals (jobs) on a single server as they arrive in order of start times. Under the assumption that the minimum and maximum job durations are unknown, they show that no online algorithm can be better than $O(\ln \Delta)$-competitive, where Δ is the ratio of the longest to the shortest job duration. Their randomized procedure, based on a converging sequence of coin-flip probabilities, achieves an $O\big((\ln \Delta)^{1+\varepsilon}\big)$ competitive ratio for any $\varepsilon > 0$. Faigle et al. [FGK96] generalized these findings by further exploring the multi-server scenario. In Chapters 3 and 4, we study similar online scheduling problems with multiple servers: in the former, intervals have a fixed duration, while in the latter, the intervals' durations vary and are job-dependent. Unlike in [LT94] and [FGK96], our setting assumes that the knowledge of the upper/lower bounds for the interval sizes are known to the algorithm.

[GL16] studied a problem called *reserve driver scheduling*, which is similar to online scheduling and closely related to our `kRental-variable` problem. In their setting, each resource (i.e., driver) is only available during specific time intervals. They propose a randomized algorithm that is $O(\ln(\Delta))$-competitive, although the constant factor in their competitive ratio is significantly larger than that achieved in our results. Similarly, [GG20] investigated a related problem, referred to as the *online reservation problem*, and proposed a static pricing algorithm with a performance guarantee of $3 \cdot \big(1 + \ln(\Delta)\big)$. In Chapter 4, where we study a related problem, we design a dynamic pricing algorithm that not only obtains a tighter performance guarantee but also employs simpler techniques compared to those in [GG20].

Online bipartite matching (`OBM`). The `OBM` problem was introduced by the seminal work of [KVV90]. `OBM` extends the online selection problem to multiple resource types, each with multiple identical copies. Over the years, different variants of `OBM` have been explored. One such instance is the online edge-weighted matching problem, which initially arose in the context of the Ad assignment problem [FKMP09]. Achieving bounded competitive ratios for the Ad assignment problem typically relies on the free disposal assumption, which may not hold in general.

[ZCL08] and [SYH$^+$22] studied online edge-weighted matching under a bounded valuation assumption (which we also adopt in this thesis) in the large-supply regime, recasting it as an online multiple knapsack problem. They develop dynamic pricing algorithms with order-optimal competitive ratios. Subsequently, [MSL20] refined these pricing function designs, achieving state-of-the-art results that attain the exact optimal competitive ratio in the large-supply setting and an asymptotically optimal ratio in the finite-supply regime.

Online matching and assortment with reusable resources. [DFNS22] studied an online bipartite matching problem where resources are reusable. This setting generalizes the online matching problem introduced by [KVV90] by allowing each matched offline node to become available again after d time units. They develop a variant of the ranking algorithm, referred to as *periodic re-ranking*, which applies the well-known ranking strategy from [KVV90] every d time steps.

[SYH$^+$23] investigated the online knapsack problem with departures, which extends the standard online knapsack problem to a reusable setting. Their formulation represents another variant of online matching with reusability, where each online node has a node-dependent rental duration for different offline nodes, and the algorithm collects a per-unit-time reward by matching an online node to an offline node. They proposed a dynamic threshold algorithm that achieves an order-optimal competitive ratio under the assumption that the knapsack capacity is large relative to the items' sizes. Similarly, [GI02] study a scheduling problem for continuous resources.

The `Matching-RR` problem studied in Chapter 6 is a more general version of the problem examined in [SYH$^+$23]. In our formulation, we do not impose any assumptions on the arrival times of buyers; in contrast, [SYH$^+$23] restricts online arrivals to occur only at discrete time steps. Furthermore, our model does not assume that item sizes are infinitesimally small—a condition that, in effect, presumes an inventory size asymptotically converging to infinity. This generalization enables us to capture a wider range of practical scenarios in online resource allocation.

[HC23] studied an online assortment problem in which each buyer belongs to a discrete set of types, and upon a buyer's arrival, the buyer's type is revealed to the decision maker. Each buyer's rental duration depends on its type and can vary across the units of the resource. This problem is also a special case of the problem in Chapter 6. When the number of buyer types M approaches infinity and the inventory size for each resource also approaches infinity, and their algorithm achieves a $4\ln\left(\frac{d_{\max}}{d_{\min}}\right)$-competitive ratio, where $d_{\max}$ ($d_{\min}$) is the maximum (minimum) duration that a buyer can request to rent a unit of a resource type.

Furthermore, a broader list of works in the revenue management literature and stochastic optimization has studied online resource allocation with reusable resources under stochastic assumptions. For instance, [FNS19] examined an online assortment problem with reusable resources, where the value obtained by allocating different resources is fixed and the rental durations of buyers are either deterministically resource-dependent or drawn from a known distribution that depends on the resource (but not on the buyer). They show that a load-balancing algorithm, similar to the one proposed by [GNR14] for online assortment with non-reusable resources, achieves $(1-1/e)$-competitiveness when the minimum inventory approaches infinity for the deterministic-duration case, and $\big(1-1/e\big)^2$ for the i.i.d. distribution scenario.

Moreover, [GGI+22] studied the same problem and demonstrated that a myopic policy attains a competitive ratio of 1/2 using a coupling technique. [GIU0] investigate both online matching and assortment planning with customer-independent usage durations, proposing a policy featuring a fluid update of capacity. Employing a novel LP-free certificate approach, they show that this policy is asymptotically optimal, achieving a competitive ratio of $\frac{e}{e-1}$.

Online rounding: Creating correlation to round fractional solutions. Recent work in computer science and operations research has highlighted the effectiveness of online rounding frameworks [FHTZ22, Ma24]. Such frameworks typically follow a *relax-and-round* approach: first, they employ an online primal-dual method to generate fractional solutions; then, they apply an online rounding scheme to convert these fractional solutions into integral decisions. While methods for generating competitive fractional solutions, such as the online primal-dual approach, are generally well-understood, the online rounding phase remains nontrivial. A crucial aspect of these rounding schemes is establishing correlation with previously made allocation decisions.

Recent years have witnessed significant progress in developing innovative online rounding schemes. For instance, [FHTZ22] propose a subroutine termed Online Correlated Selection (OCS), which introduces negative correlation across selected pairs, leading to improved competitive ratios compared to classical greedy algorithms, which achieve only a $\frac{1}{2}$-competitive ratio. Subsequently, [DFNS22] extend this concept to settings involving reusable resources and fixed rental durations by developing the Online Correlated Rental (OCR) method. Additionally, [HZZ24] adapt similar correlation-based techniques to the Adwords problem.

Within the context of stochastic optimization, several studies (e.g., [FSZ21, Ala14]) have developed analogous rounding methods, including online contention resolution schemes, demonstrating their effectiveness in applications such as prophet inequalities and online stochastic matching.

Some other related techniques: OPD, R-OPD, and LP-free Certificate. In this thesis, we employ multiple approaches to establish the performance guarantees of our algorithms. One method we use in Chapter 3 is the online primal-dual (OPD) approach, based on the work of [BN09], which designs an OPD framework for a general class of online packing problems.

In Chapter 5, we also make use of the randomized online primal-dual (R-OPD) method developed by [DJK13] to provide a more generalized proof of the optimality of the RANKING algorithm by [KVV90] for the online bipartite matching (`OBM`) problem. In R-OPD, the dual constraints hold only in expectation. [EFFS21] present an economics-based interpretation of R-OPD, extending the RANKING analysis for online bipartite matching to (i) resources with multiple copies and (ii) valuations that depend on both resources and buyers.

Additionally, in Chapters 4 and 6, we adopt the LP-free certificate approach proposed by [GIU0], which formulates a system of linear constraints whose feasibility certifies a lower bound on the competitive ratio. Departing from the conventional OPD framework, this LP-free analysis works with a linear system directly tied to the actions of the clairvoyant algorithm against which we compare. This LP-free framework has influenced later works such as [AS23], which also employ the LP-free technique to establish performance guarantees in their respective domains.

1.3 Contributions

In this thesis, we develop new algorithms for the problems discussed in Section 1.1 and analyze their theoretical performance guarantees through competitive analysis. We also establish lower bounds on the competitive ratios of online algorithms by designing hard instances for each problem and analyzing the behavior of any online algorithm on these instances. By comparing these lower bounds with our algorithmic performance, we demonstrate the effectiveness of our designs. In the following, we discuss the contributions made in each chapter independently.

Chapter 2 The primary contribution of this chapter is the development of novel posted price mechanisms employing randomized dynamic pricing schemes that extend the results in [BGMS11, HK19, TYBLG23, SJTB24]. The proposed scheme, r-Dynamic-cost (Algorithm 1), sequentially updates the price of each resource as new units are produced and sold. Specifically, as the marginal production cost increases with each additional unit, r-Dynamic-cost employs different pricing function to independently randomize the price for each unit. We also derive a new lower bound on the performance of every online algorithm.

The key technical component in deriving the aforementioned lower and upper bounds is a new *representative function*-based approach, which models the dynamics of any randomized online algorithm using a sequence of k probability functions, $\{\psi_i\}_{i\in[k]}$. We design a family of hard instances and characterize the performance of any α-competitive algorithm on these instances via a set of differential equations involving $\{\psi_i\}_{i\in[k]}$. To determine the tightest lower bound, we compute the minimum α for which these equations admit a feasible solution, namely valid probability functions $\{\psi_i\}_{i\in[k]}$. By reverse engineering these equations, we derive inverse probability functions, $\{\phi_i\}_{i\in[k]}$, for pricing each unit, which in turn leads to design of r-Dynamic-cost (Algorithm 1) in this chapter. Unlike the classic method based on Yao's Minmax principle, the representative function-based approach is not only more general and applicable to more online resource allocation, but also provides guidance for designing the pricing functions.

Chapter 3 In this chapter, motivated by works on online resource allocation with reusable resources (e.g., [DFNS22, LT94, GG20]), we introduce the problem `kRental-fixed`, which, unlike the aforementioned works, focuses on a single resource type with buyers making fixed rental duration requests for each unit of the resource. This simpler problem provides deeper insight into resource allocation in reusable settings.

We then introduce a novel online algorithm, rDynamic-fixed (Algorithm 2), which employs a dynamic pricing function to generate a fractional solution that is subsequently rounded using a novel online rounding procedure called (k, d)-oca. The (k, d)-oca procedure not only rounds these fractional allocations but also schedules the rental requests from different buyers, thereby maximizing the rental duration of the available resource units. Furthermore, we derive a lower bound for the performance guarantee of any online algorithm using the representative function approach similar to that in Chapter 3, and we prove that the competitive ratio of rDynamic-fixed meets this lower bound, thereby establishing its optimality.

From a technical perspective, the 1-(k, d)-oca approach (Algorithm 3) is a novel online rounding scheme that, at each time step, establishes correlations with previous decisions to efficiently round the fractional solution obtained in the preceding step. This new correlation scheme is lossless in the sense that, as long as the cumulative fractional allocations from previous time steps—whose corresponding rental intervals extend into the current time step—remain below k (the inventory size), the procedure can round any fractional solution so that a unit of the resource is allocated with a probability equal to the fractional value. We believe that the (k, d)-oca rounding scheme is applicable to other settings and will be of independent interest.

Chapter 4 In this chapter, we design a novel algorithm that achieves tighter performance guarantees than the algorithms proposed in [GG20, GL16] for similar problems following different design. We introduce a new algorithm called rDynamic-variable (Algorithm 4) that employs a dynamic pricing approach to generate a fractional solution, which is subsequently rounded using a new online correlation scheme developed in this work. Using the "representative function" approach [SJTB24], we derive a lower bound for the competitive ratio of online algorithms and argue that rDynamic-variable is order-optimal with respect to the fluctuation ratio corresponding to the uncertainty measure in the `kRental-variable` problem discussed in this chapter.

From a technical standpoint, we derive a system of differential inequalities that the pricing function of rDynamic-variable must satisfy for a desired level of competitiveness, using the LP-free certificate approach of [GIU0]. This system, parameterized by the achievable competitive ratio, guides us in designing the pricing function with the lowest possible competitive ratio. By approximating these inequalities via numerical methods, we formulate a finite-variable linear program whose solution yields a pricing function design matching the LP's objective value. Finally, numerical LP solvers are employed to obtain pricing functions that ensure strong performance for rDynamic-variable.

Chapter 5 Previous research has thoroughly investigated posted price mechanisms for adversarial online matching (e.g., [ZLW17, MSL20]), employing dynamic pricing that adjusts prices on a per-buyer basis, effectively engaging in price discrimination. In this chapter, we present randomized static pricing algorithms for the online edge-weighted matching (`Matching-BW`) problem and demonstrate that they achieve the optimal competitive ratios among static pricing strategies. Moreover, we derive a lower bound for the competitive ratio that is tighter than that of [MSL20], showing that not only does our static pricing algorithm reach this bound, but the previously developed dynamic pricing methods are also optimal among all online algorithms. These findings confirm that it is feasible to design non-price-discriminatory algorithms without any loss of efficiency.

Technically, we propose new approaches to analyze both the upper bound of Algorithm 5 and the lower bound of this problem. Our competitive analysis of the static pricing algorithms is grounded in an economics-inspired approach within the posted pricing mechanism framework. This approach generalizes the economics-based analysis of RANKING for online bipartite matching [EFFS21] by additionally taking into account that (i) each resource is available in multiple copies and (ii) valuations depend on both the resource and the buyer. Furthermore, our lower bound proofs are based on the representative function–based approach used in the previous chapter, which enables us to derive the optimal online algorithm over a newly designed set of hard instances.

Chapter 6 In this chapter, we introduce a new problem, `Matching-RR`, motivated by previous works [SYH+23, LT94]. In this setting, buyers may arrive at any point during the horizon, and the rental durations for each unit of the resource are buyer-dependent, with buyers possessing distinct per-unit-time valuations for renting each resource type. To address the uncertainties inherent in `Matching-RR`, we develop a new randomized algorithm that leverages two sources of randomization. First, a static pricing scheme is employed to manage uncertainty in buyers' per-unit-time valuations across different resources. Second, a randomized dynamic pricing function handles uncertainty regarding buyers' rental durations. We derive a new lower bound for the problem and demonstrate that our algorithm is order-optimal with respect to the two primary fluctuation ratios corresponding to these uncertainties.

From a technical perspective, we adopt a randomized version of the LP-free certificate approach from [GIU0]—in which the constraints hold only in expectation—to establish the performance guarantee of Algorithm 6 presented in this chapter. Additionally, we employ a representative-function

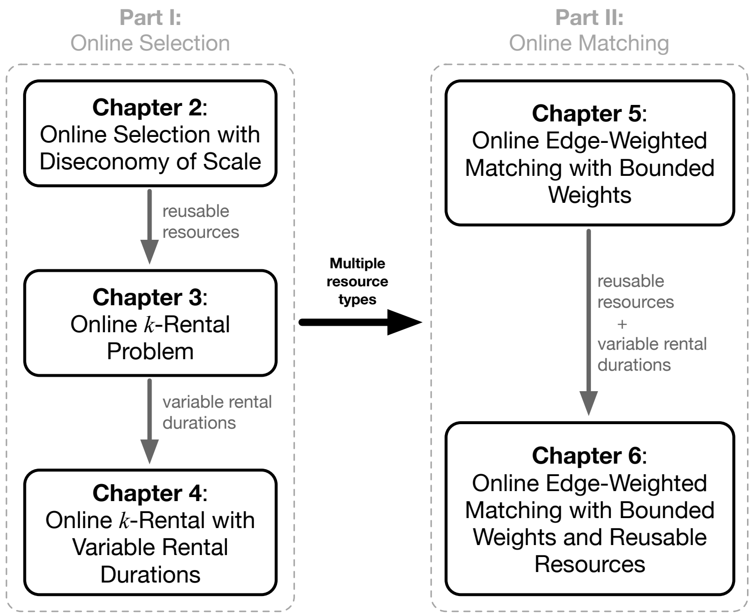


Figure 1.1: Structure of the thesis and connections between problems studied by different chapters.

based approach to derive a lower bound on the performance guarantee achievable by any online algorithm for the `Matching-RR` problem.

1.4 Structure of the Thesis

This thesis is structured into two main parts, each focusing on variants of adversarial online allocation problems.

In Part I, we study different variants of the adversarial online selection problem, where the seller has a single resource type. This part consists of three chapters. Chapter 2 examines online selection with diseconomies of scale (`kSelection-cost`), where the seller faces scenarios with increasing marginal production costs. Chapters 3 and 4 study the online k-rental problem, differentiating between fixed (i.e., `kRental-fixed`) and variable rental durations (i.e., `kRental-variable`), where the decision maker must address the reusability of the resource.

Part II shifts the focus to online matching problems, where the seller manages multiple resource types and faces matching constraints. Chapter 5 discusses the online edge-weighted matching (`Matching-BW`) problem, while Chapter 6 investigates online edge-weighted matching with reusable resources (`Matching-RR`).

The thesis concludes in Chapter 7 with a discussion of the results, key contributions, and potential directions for future research.

Part I

ONLINE SELECTION

Chapter 2

Online Selection with Diseconomies of Scale

In this chapter, we investigate the online selection problem with diseconomies of scale (`kSelection-cost`), where a seller must allocate a limited number of resources immediately and irrevocably to sequentially arriving buyers with distinct valuations, under conditions of increasing marginal production costs. We first derive a novel, tight lower bound on the competitive ratio achievable by any online algorithm for this problem. Leveraging insights from this lower bound, we propose a randomized dynamic pricing algorithm, r-Dynamic-cost, which adjusts resource prices dynamically based on marginal production costs and remaining inventory. We show that this approach is optimal for small inventory sizes and consistently outperforms existing algorithms in larger-scale scenarios, achieving an optimal competitive ratio in asymptotically large inventory settings.

2.1 Problem Statement and Assumptions

We formally define online k-selection with diseconomies of scale (`kSelection-cost`) as follows. Consider an online market operating under posted price mechanisms. On the supply side, a seller can produce a total of k units of a resource, with increasing (or at least non-decreasing) marginal production costs. Let $\boldsymbol{c} := \{c_i\}_{\forall i\in[k]}$ represent the *marginal production cost*, where c_i denotes the cost of producing the i-th unit, and $c_1 \leq c_2 \leq \cdots \leq c_k$. Define $f(i) = \sum_{j=1}^{i} c_j$ as the *cumulative production cost* of the first i units. On the demand side, N buyers arrive sequentially, each demanding one unit of the resource. Let v_n denote the private valuation of the n-th buyer. Once buyer n arrives, a price p_n is posted, and then the buyer decides to accept the price and make a purchase if a non-negative utility is gained $v_n - p_n \geq 0$, and reject it otherwise.

Let $x_n \in \{0, 1\}$ represent the decision of buyer n, where $x_n = 1$ indicates a purchase and $x_n = 0$

otherwise. Then buyer n obtains a utility $(v_n - p_n)x_n$ and the seller collects a total revenue of $\sum_{n\in[N]} p_n x_n - f(\sum_{n\in[N]} x_n)$ from all buyers. The goal of the online market is to determine the posted prices $\{p_n\}_{\forall n\in[N]}$ to maximize the social welfare, which is the sum of utilities of all the buyers and the revenue of the producer, i.e., $\sum_{n\in[N]} x_n \cdot (v_n - p_n) + \sum_{n\in[N]} x_n \cdot p_n - f(\sum_{n\in[N]} x_n) = \sum_{n\in[N]} v_n x_n - f(\sum_{n\in[N]} x_n)$.

Let $\mathcal{I} = \{v_1, \cdots, v_N\}$ denote an arrival instance of buyers. An optimal offline clairvoyant algorithm can obtain the optimal social welfare $\mathtt{OPT}(\mathcal{I})$ by solving the following optimization problem

$$\mathtt{OPT}(\mathcal{I}) = \max_{x_n\in\{0,1\}} \sum\nolimits_{n\in[N]} v_n x_n - f\left(\sum\nolimits_{n\in[N]} x_n\right), \qquad \text{s.t. } \sum\nolimits_{n\in[N]} x_n \le k.$$

However, in the online market, the posted price p_n is determined without knowing the valuations of future buyers $\{v_\tau\}_{\tau>n}$. We aim to design an online mechanism to determine the posted prices such that the social welfare achieved by the online mechanism, denoted by $\mathtt{ALG}(\mathcal{I})$, is competitive compared to $\mathtt{OPT}(\mathcal{I})$, where the competitive ratio of an online algorithm is defined according to Eq. (1.1).

To attain a bounded competitive ratio, we consider a constrained adversary model [JLTZ21b, TYBLG23], where the buyers' valuations are assumed to be bounded.

Assumption 2.1. *Buyers' valuations are bounded in* $[v_{\min}, v_{\max}]$*, i.e.,* $v_n \in [v_{\min}, v_{\max}], \forall n \in [N]$.

The interval $[v_{\min}, v_{\max}]$ can be considered as the prediction interval that covers the valuations of all buyers [JLTZ21b], and is known to the online algorithm. As shown in [TYBLG23], the competitive analysis of online algorithms for `kSelection-cost` depends on the relationship between buyers' valuations and the production cost function. For simplicity, we focus on the case where the production cost is always smaller than the buyer's valuation ($c_k < v_{\min}$) and derive lower and upper bounds in Sections 2.2 and 2.3, respectively. In Appendices A.9 and A.10, we show that this assumption is without loss of generality, as our results extend naturally to the general case.

2.2 Lower Bound for `kSelection-cost`: Hardness of Allocation with Diseconomies of Scale

We first derive a tight lower bound for `kSelection-cost`, which motivates the design of r-Dynamic-cost (Algorithm 1) in Section 2.3.

2.2.1 Lower Bound $\alpha^*_{\mathcal{S}}(k)$

Theorem 2.1 below formally states the lower bound $\alpha^*_{\mathcal{S}}(k)$ for the competitive ratio of any online algorithm for `kSelection-cost`.

Theorem 2.1 (Lower Bound)**.** *Given $\mathcal{S} = \{v_{\min}, v_{\max}, f\}$ for the `kSelection-cost` problem with $k \geq 1$, no online algorithm, including those with randomization, can achieve a competitive ratio smaller than $\alpha^*_{\mathcal{S}}(k)$, where $\alpha^*_{\mathcal{S}}(k)$ is the solution to the following equation of α:*

$$\begin{aligned} v_{\max} = \left(v_{\min} - c_{\underline{k}}\right) \cdot e^{\frac{\alpha}{k}\cdot(k+1-\underline{k}-\xi)} + c_{\underline{k}} \cdot e^{\frac{\alpha}{k}\cdot(k-\underline{k})} \\ + c_{\underline{k}+1} \cdot \left(1 - e^{\frac{\alpha}{k}}\right) \cdot e^{\frac{\alpha}{k}\cdot(k-1-\underline{k})} + \cdots + c_k \cdot \left(1 - e^{\frac{\alpha}{k}}\right). \end{aligned} \tag{2.1}$$

In Eq. (2.1), $\underline{k} \in [k]$ denotes the smallest natural number such that

$$\sum\nolimits_{i=1}^{\underline{k}} (v_{\min} - c_i) \geq \frac{1}{\alpha} \cdot \left(k v_{\min} - \sum\nolimits_{i=1}^{k} c_i\right), \tag{2.2}$$

and $\xi \in (0,1]$ denotes the unique solution to the following equation

$$\xi = \frac{\frac{1}{\alpha} \cdot (k v_{\min} - \sum_{i=1}^{k} c_i) - \sum_{i=1}^{\underline{k}-1} (v_{\min} - c_i)}{v_{\min} - c_{\underline{k}}}. \tag{2.3}$$

Theorem 2.1 is our main result concerning the hardness of `kSelection-cost`. To prove Theorem 2.1, a key step is to establish a set of necessary conditions that any α-competitive online algorithm must satisfy. A formal proof will be provided in Section 2.2.3. Below, we offer several remarks to clarify the key intuitions.

- By the definition of $\underline{k}$ in Eq. (2.2), $\underline{k}$ represents the minimum number of units that any α-competitive deterministic algorithm, denoted by $\mathtt{ALG_d}$, must sell when faced with an arrival instance of k identical buyers with valuation $v_{\min}$, denoted by $\mathcal{I}^{(v_{\min})}_{iden} = \{v_{\min}, \cdots, v_{\min}\}$. Under the instance $\mathcal{I}^{(v_{\min})}_{iden}$, the maximum social welfare achievable by the offline optimal algorithm is $k v_{\min} - \sum_{i=1}^{k} c_i$. Therefore, $\mathtt{ALG_d}$ must sell at least $\underline{k}$ units to ensure α-competitiveness, implying that $\underline{k}$ is well-defined for all values of $\alpha \geq 1$.

- Eq. (2.3) demonstrates that ξ is defined as the fraction of the $\underline{k}$-th unit required to make Eq. (2.2) binding. We argue that $\xi \in (0,1]$ is well-defined and always exists as long as there is an α-competitive randomized algorithm, denoted as $\mathtt{ALG_r}$. Specifically, if a randomized algorithm $\mathtt{ALG_r}$ is run on the same instance $\mathcal{I}^{(v_{\min})}_{iden}$, $\mathtt{ALG_r}$ must sell at least $\underline{k} - 1$ units plus a fraction ξ of the $\underline{k}$-th unit of the resource, in expectation.

- Note that, in general, a closed-form expression for the lower bound $\alpha^*_{\mathcal{S}}(k)$ cannot be derived. This is expected due to the arbitrary nature of the sequence of marginal production costs.

However, because of the monotonicity of $\underline{k}$, ξ, and the right-hand side of Eq. (2.1) with respect to α, $\alpha^*_{\mathcal{S}}(k)$ can be easily computed by solving Eq. (2.1) numerically using binary search.

In the next subsection, we construct a family of hard instances and introduce a novel representative function-based approach to derive a system of differential equations, which are crucial to proving the lower bound result in Theorem 2.1.

2.2.2 Representing Worst-Case Performance by (Probabilistic) Allocation Functions

We introduce a family of hard instances based on the instance $\mathcal{I}^{(\epsilon)}$ defined as follows.

Definition 2.1 (Instance $\mathcal{I}^{(\epsilon)}$)**.** *For any given value of $\epsilon > 0$, the instance $\mathcal{I}^{(\epsilon)}$ begins with k identical buyers, each having a valuation of $v_{\min}$ during the initial stage. This is followed by a series of stages, each consisting of k identical buyers, with valuations incrementally increasing by ϵ, starting from $v_{\min} + \epsilon$ and reaching the upper bound $v_{\min} + \lfloor \frac{v_{\max} - v_{\min}}{\epsilon} \rfloor \cdot \epsilon$. The instance $\mathcal{I}^{(\epsilon)}$ is mathematically defined as:*

$$\Bigg\{ \underbrace{v_{\min}, \ldots, v_{\min}}_{k \text{ buyers}}, \underbrace{v_{\min} + \epsilon, \ldots, v_{\min} + \epsilon}_{k \text{ buyers}}, \ldots, \underbrace{v_{\min} + j \cdot \epsilon, \ldots, v_{\min} + j \cdot \epsilon}_{k \text{ buyers in stage } v_{\min} + j \cdot \epsilon},$$
$$\ldots, \underbrace{v_{\min} + \left\lfloor \frac{v_{\max} - v_{\min}}{\epsilon} \right\rfloor \cdot \epsilon, \ldots, v_{\min} + \left\lfloor \frac{v_{\max} - v_{\min}}{\epsilon} \right\rfloor \cdot \epsilon}_{k \text{ buyers}} \Bigg\},$$

where j ranges from 1 to $\lfloor (v_{\max} - v_{\min})/\epsilon \rfloor$. Furthmore, let us define the set $V^{(\epsilon)} = \{v_{\min}, v_{\min} + \epsilon, \ldots, v_{\min} + \lfloor (v_{\max} - v_{\min})/\epsilon \rfloor \cdot \epsilon\}$ to contain all the possible valuations that buyers in the instance $\mathcal{I}^{(\epsilon)}$ may possess.

We refer to the k buyers with valuation $v \in V^{(\epsilon)}$ as *stage-v* arrivals in $\mathcal{I}^{(\epsilon)}$. For any $v \in V^{(\epsilon)}$, let $\mathcal{I}^{(\epsilon)}_v$ denote all the buyers in $\mathcal{I}^{(\epsilon)}$ from the beginning up to *stage-v*. For instance, if $v = v_{\min} + 2\epsilon$, then $\mathcal{I}^{(\epsilon)}_v$ includes the first $3k$ buyers in $\mathcal{I}^{(\epsilon)}$ with valuations $v_{\min}$, $v_{\min} + \epsilon$, and $v_{\min} + 2\epsilon$. Due to the online nature of the problem, we emphasize that $\mathcal{I}^{(\epsilon)}$ may terminate at any stage v. In other words, there exists a family of hard instances, $\{\mathcal{I}^{(\epsilon)}_v\}_{\forall v \in V^{(\epsilon)}}$, induced by $\mathcal{I}^{(\epsilon)}$. Here, $\mathcal{I}^{(\epsilon)}_v$ denotes the arrival instance of $\mathcal{I}^{(\epsilon)}$ that terminates at stage-v. Henceforth, we will use "*instance $\mathcal{I}^{(\epsilon)}_v$*" and "*instance $\mathcal{I}^{(\epsilon)}$ by the end of stage-v*" interchangeably.

Given any α-competitive algorithm `ALG`, an arbitrary instance from $\{\mathcal{I}^{(\epsilon)}_v\}_{\forall v \in V^{(\epsilon)}}$ may be the one that `ALG` processes. Thus, for any $v \in V^{(\epsilon)}$, by the end of stage-v of $\mathcal{I}^{(\epsilon)}$, `ALG` must achieve at least a $1/\alpha$ fraction of the optimal social welfare, $kv - \sum_{i=1}^{k} c_i$, which is attained by rejecting all previous buyers except for the last k buyers with valuation v. Consequently, an α-competitive algorithm must ensure

$$\texttt{ALG}\left(\mathcal{I}_v^{(\epsilon)}\right) \geq \frac{1}{\alpha} \cdot \left(kv - \sum\nolimits_{i=1}^{k} c_i\right), \quad \forall v \in V^{(\epsilon)}, \tag{2.4}$$

where $\texttt{ALG}(\mathcal{I}_v^{(\epsilon)})$ denotes the *expected* performance of $\texttt{ALG}$ under the instance $\mathcal{I}_v^{(\epsilon)}$.

Representing $\texttt{ALG}(\mathcal{I}_v^{(\epsilon)})$ by Allocation Functions. For any randomized algorithm, we define $k+1$ states, $\{q_i\}_{\forall i \in \{0,\cdots,k\}}$, which represent the allocation behavior of the online algorithm at any stage of instance $\mathcal{I}^{(\epsilon)}$, as follows:

- State q_0 corresponds to the situation where the online algorithm has not allocated any units.
- For all $i \in [k]$, state q_i represents that the online algorithm has allocated *at least* i units of the resource.

For all $v \in V^{(\epsilon)}$ and $i \in \{0, \cdots, k\}$, we define $\Psi_i(v) : V^{(\epsilon)} \to \{0,1\}$ such that $\Psi_i(v) = 1$ if the algorithm is in state q_i after processing all the buyers in $\mathcal{I}_v^{(\epsilon)}$, and $\Psi_i(v) = 0$ otherwise. Specifically, $\Psi_i(v) = 1$ if the online algorithm allocates at least i units of the resource at the end of stage v in $\mathcal{I}^{(\epsilon)}$, which occurs with some probability depending on the algorithm's randomness. Since the instance $\mathcal{I}^{(\epsilon)}$ is deterministically defined, $\Psi_i(v)$ is a binary random variable whose distribution depends solely on the algorithm's randomness. This leads to the definition of $\boldsymbol{\psi} = \{\psi_i\}_{\forall i \in [k]}$ below.

Definition 2.2 (Allocation Functions)**.** *For any randomized online algorithm, let $\boldsymbol{\psi} = \{\psi_i\}_{\forall i \in [k]}$ and $\psi_i : V^{(\epsilon)} \to [0,1]$ represent the functions where $\psi_i(v) = \mathbb{E}[\Psi_i(v)]$, with the expectation taken over the randomness of the algorithm.*

Based on the definition above, we have $\psi_i(v) = \Pr(\Psi_i(v) = 1)$, where $\Psi_i(v) = 1$ indicates that the algorithm is in state q_i (i.e., at least i units of the resource have been allocated) after processing all buyers in $\mathcal{I}_v^{(\epsilon)}$ (i.e., by the end of stage v of instance $\mathcal{I}^{(\epsilon)}$). In this context, $\psi_i(v)$ represents the probability that the online algorithm has allocated at least i units of the resource by the end of stage v in instance $\mathcal{I}^{(\epsilon)}$. Therefore, the term *probabilistic allocation functions* is used or simply *allocation functions* for brevity. We show that $\psi_i(v)$ is monotonic in $i \in [k]$.

Lemma 2.1 (Monotonicity)**.** *For any randomized online algorithm, $\psi_i(v) \geq \psi_{i+1}(v)$ holds for all $i \in [k]$ and $v \in [v_{\min}, v_{\max}]$.*

The proof of the above lemma is given in Appendix A.1. Lemma 2.1 implies that it suffices to focus on randomized algorithms whose allocation functions are from the following set

$$\Omega = \left\{\boldsymbol{\psi} \middle| \psi_i(v) \in [0,1], \psi_i(v) \geq \psi_{i+1}(v), \psi_i(v) \leq \psi_i(v'), \forall i \in [k], v, v' \in V^{(\epsilon)}, \text{ and } v < v'\right\}.$$

Next, we analyze how the allocation level of an α-competitive algorithm should evolve as new buyers with higher valuations arrive in $\mathcal{I}^{(\epsilon)}$. We argue that the expected performance of any online algorithm under the instance $\mathcal{I}^{(\epsilon)}$ can be fully represented by the k allocation functions $\{\psi_i(v)\}_{\forall i\in[k]}$. Let $\texttt{ALG}(\mathcal{I}_v^{(\epsilon)})$ denote the expected objective value of the algorithm under instance $\mathcal{I}_v^{(\epsilon)}$. Then $\texttt{ALG}(\mathcal{I}_v^{(\epsilon)})$ can be framed using $\boldsymbol{\psi} = \{\psi_i(v)\}_{\forall i\in[k]}$ as follows.

Proposition 2.1 (Representation based on $\boldsymbol{\psi}$)**.** *For any randomized algorithm* ALG *under the family of hard instances* $\{\mathcal{I}_v^{(\epsilon)}\}_{\forall v\in V^{(\epsilon)}}$*, its expected performance can be represented by its allocation functions* $\{\psi_i(v)\}_{\forall i\in[k]} \in \Omega$ *as follows:*

$$\mathtt{ALG}\left(\mathcal{I}_{v_{\min}}^{(\epsilon)}\right) = \sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_i),$$

$$\mathtt{ALG}\left(\mathcal{I}_{v_{\min}+j\cdot\epsilon}^{(\epsilon)}\right) = \mathtt{ALG}\left(\mathcal{I}_{v_{\min}}^{(\epsilon)}\right) + \sum_{i=1}^{k}\sum_{m=1}^{j} \Big[(v_{\min} + m\cdot\epsilon - c_i)\cdot\Big(\psi_i(v_{\min}+m\cdot\epsilon) - \psi_i(v_{\min}+(m-1)\cdot\epsilon)\Big)\Big], \quad \forall j = 1,2,\ldots,\lfloor\frac{v_{\max}-v_{\min}}{\epsilon}\rfloor.$$

The above proposition relates the expected performance of an online algorithm to the set of allocation functions $\{\psi_i(v)\}_{\forall i\in[k]}$ that capture its dynamics under hard instances $\{\mathcal{I}_v^{(\epsilon)}\}_{\forall v\in V^{(\epsilon)}}$. The detailed proof can be found in Appendix A.2.

Combining Proposition 2.1 and Eq. (2.4) gives the lemma below.

Lemma 2.2 (Necessary Conditions)**.** *If there exists an α-competitive algorithm for* OSDoS*, then there exists k allocation functions $\{\psi_i\}_{i\in[k]} \in \Omega$, where each function $\psi_i : [v_{\min}, v_{\max}] \to [0,1]$ is continuous within its range and also satisfies the following equation:*

$$\sum_{i=1}^{k} \psi_i(v_{\min})\cdot(v_{\min}-c_i) + \sum_{i=1}^{k}\int_{\eta=v_{\min}}^{v} (\eta - c_i)d\psi_i(\eta) \geq \frac{1}{\alpha}\cdot\left(kv - \sum_{i=1}^{k} c_i\right), \quad \forall v\in[v_{\min}, v_{\max}]. \tag{2.5}$$

The above result is derived based on the family of instances $\{\mathcal{I}_v^{(\epsilon)}\}_{\forall v\in V^{(\epsilon)}}$ when ϵ approaches to zero. The proof is given in Appendix A.3. The lemma above provides a set of necessary conditions for the allocation functions $\{\psi_i\}_{\forall i\in[k]}$ induced by any α-competitive algorithm. Therefore, determining a tight lower bound for `kSelection-cost` is equivalent to finding the lowest α such that there exists a set of allocation functions in Ω that satisfy Eq. (2.5).

2.2.3 Proof of Theorem 2.1

We now move on to prove Theorem 2.1. Based on the necessary conditions in Lemma 2.2, the lower bound can be defined as

$$\alpha^*_{\mathcal{S}}(k) = \inf\Big\{\alpha \geq 1 \;\Big|\; \text{there exist a set of } k \\ \text{allocation functions } \{\psi_i(v)\}_{\forall i\in[k]} \in \Omega \text{ that satisfy Eq. (2.5)}\Big\}.$$

Next, we show that it is possible to find a tight design of $\{\psi_i\}_{\forall i\in[k]}$ that satisfies the necessary conditions in Eq. (2.5) by equality, ultimately leading to Eq. (2.1) in Theorem 2.1.

For any $\alpha \geq \alpha^*_{\mathcal{S}}(k)$, let $\Gamma^{(\alpha)}$ denote the superset of the set of functions $\{\psi_i\}_{\forall i\in[k]} \in \Omega$ that satisfy Eq. (2.5). Note that $\Gamma^{(\alpha)} \subset \Omega$ holds for all $\alpha \geq \alpha^*_{\mathcal{S}}(k)$. Define $\chi^{(\alpha)}(v) : [v_{\min}, v_{\max}] \to [0, k]$ as

$$\chi^{(\alpha)}(v) = \inf\left\{\sum\nolimits_{i=1}^{k} \psi_i(v) \;\Big|\; \{\psi_i(v)\}_{\forall i\in[k]} \in \Gamma^{(\alpha)}\right\}. \tag{2.6}$$

Based on the definition of $\chi^{(\alpha)}$, we construct a set of allocation functions $\{\psi_i^{(\alpha)}(v)\}_{\forall i\in[k]}$ as follows:

$$\psi_i^{(\alpha)}(v) = \Big(\chi^{(\alpha)}(v) - (i-1)\Big)\cdot \mathbf{1}_{\{i-1\leq\chi^{(\alpha)}(v)\leq i\}} + \mathbf{1}_{\{\chi^{(\alpha)}(v)>i\}}, \forall v \in [v_{\min}, v_{\max}], \quad \forall i \in [k], \tag{2.7}$$

where $\mathbf{1}_{\{A\}}$ is the standard indicator function, equal to 1 if A is true and 0 otherwise. In the following lemma, we argue that the set of functions $\{\psi_i^{(\alpha)}(v)\}_{\forall i\in[k]}$ is a feasible solution to Eq. (2.5) and satisfies it as an equality.

Lemma 2.3. *For any $\alpha \geq \alpha^*_{\mathcal{S}}(k)$, the functions $\{\psi_i^{(\alpha)}\}_{\forall i\in[k]}$ satisfy Eq. (2.5) as an equality.*

The detailed proof for the above lemma is in Appendix A.4. Following the definition of $\{\psi_i^{(\alpha)}(v)\}_{\forall i\in[k]}$, we observe that these functions exhibit the following property:

Lemma 2.4. *For any $i \in [k]$ and $v \in [v_{\min}, v_{\max}]$, if $\psi_i^{(\alpha)}(v) \in (0,1)$ holds, then $\psi_j^{(\alpha)}(v) = 1$ for all $j = 1, \cdots, i-1$ and $\psi_j^{(\alpha)}(v) = 0$ for all $j = i+1, \cdots, k$.*

Lemma 2.4 asserts that if the online algorithm inducing $\{\psi^{(\alpha)}\}_{\forall i}$ begins allocating unit i with some positive probability to buyers in stage-v of $\mathcal{I}^{(\epsilon)}$, then the algorithm must have already allocated all units $j < i$ with probability one to buyers arriving at or before stage-v of $\mathcal{I}^{(\epsilon)}$. Furthermore, if the algorithm has not allocated unit i with probability one by the end of stage-v, then all units $j > i$ remain in the system with probability one at the end of stage-v. Given that the marginal cost for each additional unit of resource increases, the algorithm should only produce and allocate a new unit once all previously produced units have been fully allocated.

According to Lemma 2.3, the inequality in Eq. (2.5) can be replaced with an equality. By combining Lemma 2.3 with Lemma 2.4, we conclude that there exists a unique set of functions that satisfy Eq. (2.5) as an equality and also fulfill the property stated in Lemma 2.4. Proposition 2.2 below formally states this result.

Proposition 2.2. *For any $\alpha \geq \alpha^*_{\mathcal{S}}(k)$, there exist a set of allocation functions $\{\psi_i^{(\alpha)}\}_{\forall i\in[k]} \in \Omega$ that satisfy Eq. (2.5) by equality:*

$$
\begin{aligned}
&\psi_i^{(\alpha)}(v) = 1, \quad i = 1,\dots,\underline{k}-1,\\
&\psi_{\underline{k}}^{(\alpha)}(v) = \begin{cases} \xi + \frac{k}{\alpha}\cdot \ln\left(\frac{v-c_{\underline{k}}}{v_{\min}-c_{\underline{k}}}\right) & v\in[v_{\min}, u_{\underline{k}}],\\ 1 & v > u_{\underline{k}},\end{cases}\\
&\psi_i^{(\alpha)}(v) = \begin{cases} 0 & v\leq \ell_i,\\ \frac{k}{\alpha}\cdot\ln\left(\frac{v-c_i}{\ell_i-c_i}\right) & v\in[\ell_i,u_i],\\ 1 & v\geq u_i,\end{cases} \quad i = \underline{k}+1,\dots,k-1,\\
&\psi_k^{(\alpha)}(v) = \begin{cases} 0 & v\leq \ell_k,\\ \frac{k}{\alpha}\cdot\ln\left(\frac{v-c_k}{\ell_k-c_k}\right) & v\in[\ell_k, v_{\max}],\end{cases}
\end{aligned}
$$

where the intervals $\{[\ell_i,u_i]\}_{\forall i}$ are specified by

$$u_{\underline{k}} = \ell_{\underline{k}+1} = (v_{\min} - c_{\underline{k}})\cdot e^{(1-\xi)\cdot\frac{\alpha}{k}} + c_{\underline{k}}, \tag{2.8}$$

$$u_i = \ell_{i+1} = (\ell_i - c_i)\cdot e^{\alpha/k} + c_i, \qquad \forall i = \underline{k}+1,\dots,k. \tag{2.9}$$

Recall that the parameters $\underline{k}$ and ξ are defined in Eq. (2.2) and Eq. (2.3), respectively. Once α is given, both $\underline{k}$ and ξ can be uniquely determined. Therefore, the set of allocation functions $\{\psi_i^{(\alpha)}\}_{\forall i\in[k]}$ given in Proposition 2.2 can also be explicitly computed once α is given. The full proof of how to derive the explicit designs of $\{\psi_i^{(\alpha)}\}_{\forall i\in[k]}$ is given in Appendix A.5.

Putting together Eq. (2.8) and Eq. (2.9), we have

$$u_k = (v_{\min} - c_{\underline{k}})\cdot e^{\frac{\alpha}{k}\cdot(k+1-\underline{k}-\xi)} + c_{\underline{k}}\cdot e^{\frac{\alpha}{k}\cdot(k-\underline{k})} + c_{\underline{k}+1}\cdot(1-e^{\frac{\alpha}{k}})\cdot e^{\frac{\alpha}{k}\cdot(k-1-\underline{k})} + \cdots + c_k\cdot(1-e^{\frac{\alpha}{k}}).$$

Note that the right-hand side of the equation above is increasing in α. Therefore, as α decreases, the value of u_k also decreases and will eventually fall below $v_{\max}$ for a specific value of α. Consequently, according to the definition of $\psi_k^{(\alpha)}$ in Proposition 2.2, $\psi_k^{(\alpha)}(v_{\max})$ will exceed 1 (since $\psi_k^{(\alpha)}(v_{\max}) > \psi_k^{(\alpha)}(u_k)$, and based on Eq. (2.9), $\psi_k^{(\alpha)}(u_k)$ is equal to one). However, this will generate an infeasible allocation function $\psi_k^{(\alpha)}$, as we require that $\psi_k^{(\alpha)}(v)\leq 1$ holds for all $v\in[v_{\min},v_{\max}]$. As a result, for those values of α where $u_k < v_{\max}$, the set of k allocation functions $\{\psi_i^{(\alpha)}\}_{\forall i\in[k]}$ obtained in

Proposition 2.2 becomes infeasible, meaning that α must be less than $\alpha^*_{\mathcal{S}}(k)$. Therefore, $\alpha^*_{\mathcal{S}}(k)$ is the value of α for which $u_k = v_{\max}$, and this gives Eq. (2.1) in Theorem 2.1. Thus, we complete the proof of Theorem 2.1.

2.3 r-Dynamic-cost: A Randomized Dynamic Posted Pricing Mechanism

We propose a randomized dynamic pricing mechanism (r-Dynamic-cost), as described in Algorithm 1, to solve the `kSelection-cost` problem. Before the buyers arrive, r-Dynamic-cost samples k independent random prices $\{P_i\}_{\forall i\in[k]}$, where P_i is the price for the i-th unit of the resource. Specifically, for each unit $i \in [k]$, a random seed s_i is drawn from the uniform distribution $\mathrm{Unif}(0,1)$, and the random price is set as $P_i = \phi_i(s_i)$, where $\phi_i(s_i)$ is the *pricing function* designed for the i-th unit. r-Dynamic-cost then posts the price of the available unit with the smallest index from $\{P_i\}_{\forall i\in[k]}$ to the online arriving buyers.

For all $i \in [k]$, the pricing function $\phi_i : [0,1] \to [L_i, U_i]$ is constructed such that the k *price intervals* $\{[L_i, U_i]\}_{\forall i\in[k]}$ span the entire range of $[v_{\min}, v_{\max}]$, where $v_{\min} = L_1 \le U_1 = L_2 \le U_2 \le \cdots \le U_{k-1} = L_k \le U_k = v_{\max}$. That is, the upper boundary of ϕ_i (i.e., the maximum price of P_i) is the lower boundary of ϕ_{i+1} (i.e., the minimum price of P_{i+1}). As a result, the posted prices will always be non-decreasing (i.e., $P_1 \le P_2 \le \cdots \le P_k$), regardless of the realization of the random seeds $\{s_i\}_{\forall i\in[k]}$. This design ensures that units with higher production costs are sold at higher prices, which is consistent with the natural pricing scheme where more expensive units reflect higher production costs.

2.3.1 Theoretical Guarantee of r-Dynamic-cost

We show that by carefully designing the pricing functions, r-Dynamic-cost achieves an asymptotically optimal competitive ratio.

Theorem 2.2. *Given $\mathcal{S} = \{v_{\min}, v_{\max}, f\}$ for the `kSelection-cost` problem with $k \ge 1$, r-Dynamic-cost is $\alpha^*_{\mathcal{S}}(k) \cdot \exp(\frac{\alpha^*_{\mathcal{S}}(k)}{k})$-competitive when the pricing functions are given by*

$$\phi_i(s) = v_{\min}, \quad \forall s \in [0,1], i \in [\underline{k}^* - 1],$$

$$\phi_{\underline{k}^*}(s) = \begin{cases} v_{\min} & s \in [0, \xi^*], \\ (v_{\min} - c_{\underline{k}^*}) \cdot e^{(s-\xi^*)\cdot\alpha^*_{\mathcal{S}}(k)/k} + c_{\underline{k}^*} & s \in [\xi^*, 1], \end{cases}$$

$$\phi_i(s) = (L_i - c_i) \cdot e^{s\cdot\alpha^*_{\mathcal{S}}(k)/k} + c_i, \quad \forall s \in [0,1], i = \underline{k}^* + 1, \ldots, k,$$

Algorithm 1: Randomized Dynamic Posted Pricing Mechanism (r-Dynamic-cost) for `kSelection-cost`

Input: Pricing functions $\{\phi_i\}_{i\in[k]}$.
Initialize:
Let $\kappa_1 = 1$. ▷ Index of the unit to be sold
Sample a random seed vector $\boldsymbol{s} = \{s_i\}_{i\in[k]}$, with each $s_i \sim U(0,1)$ independently.
Set price vector $\mathbf{P} = \{P_i\}_{i\in[k]}$, where $P_i = \phi_i\big(s_i\big)$.
for *each arriving buyer* n **do**
  if $\kappa_n \leq k$ **then**
    Post the price $p_n = P_{\kappa_n}$ to buyer n.
    if *buyer* n *accepts the price* **then**
      A unit is sold; set $x_n = 1$.
    end
  end
  Update: $\kappa_{n+1} = \kappa_n + x_n$. ▷ If buyer n rejects p_n, then $x_n = 0$.
end

where $\underline{k}^*$ and ξ^* are respectively the values of $\underline{k}$ and ξ defined in Theorem 2.1, corresponding to $\alpha = \alpha^*_{\mathcal{S}}(k)$, and the price intervals $\{[L_i, U_i]\}_{\forall i\in[k]}$ are given as follows:

$$U_{\underline{k}^*} = L_{\underline{k}^*+1} = (v_{\min} - c_{\underline{k}^*}) \cdot e^{(1-\xi^*)\cdot \alpha^*_{\mathcal{S}}(k)/k} + c_{\underline{k}^*}, \tag{2.10}$$

$$U_i = L_{i+1} = (L_i - c_i) \cdot e^{\alpha^*_{\mathcal{S}}(k)/k} + c_i, \quad \forall i = \underline{k}^* + 1, \ldots, k. \tag{2.11}$$

We provide a proof of Theorem 2.2 in Appendix A.6. At a high level, the design of the pricing functions $\{\phi_i(s)\}_{\forall i\in[k]}$ is inspired by the dynamics of an $\alpha^*_{\mathcal{S}}(k)$-competitive algorithm on the arrival instance $\mathcal{I}^{(\epsilon)}$ studied in the lower bound section. Essentially, the inverse of the pricing function $\phi_i(s)$, defined as $\phi_i^{-1}(v) = \sup\{s : \phi_i(s) \leq v\}$, follows the same design as $\psi_i^{(\alpha)}(v)$ in Proposition 2.2 when $\alpha = \alpha^*_{\mathcal{S}}(k)$, namely, $\psi_i^{(\alpha^*_{\mathcal{S}}(k))}(v) = \sup\{s : \phi_i(s) \leq v\}$.

Due to the arbitrary nature of the cost function f, neither our work nor [TYBLG23] can derive a closed-form expression for the competitive ratio, thereby preventing a direct comparison between our r-Dynamic-cost algorithm and the deterministic dynamic pricing mechanism (d-Dynamic-cost) in [TYBLG23]. In Figure 2.1, we compare the asymptotic performance of r-Dynamic-cost with d-Dynamic-cost from [TYBLG23] and the randomized static pricing mechanism (r-Static-cost) in [SJTB24]. The results demonstrate that r-Dynamic-cost significantly outperforms both d-Dynamic-cost and r-Static-cost, converging more rapidly to the lower bound as $k \to \infty$.

Asymptotic optimality of r-Dynamic-cost in general settings. Previous studies (e.g., [HK19, TYBLG23]) have shown that $\alpha^*_{\mathcal{S}}(k)$ remains bounded by a constant as $k \to \infty$. Thus,

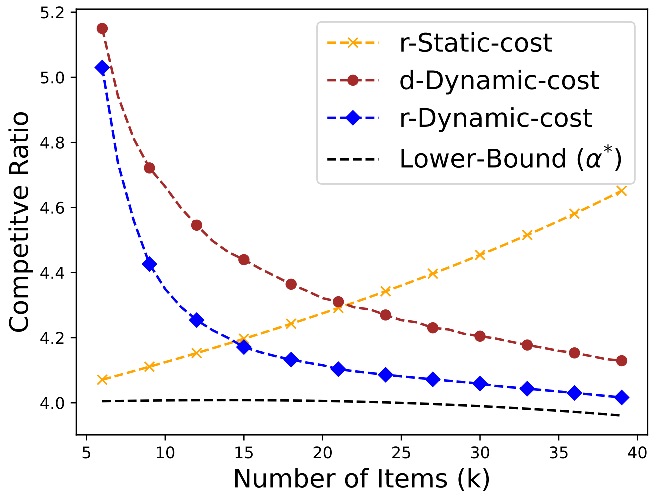


Figure 2.1: The blue curve (i.e., r-Dynamic-cost) represents the competitive ratio of Algorithm 1, which employs randomized dynamic pricing. The red curve (i.e., d-Dynamic-cost) and the yellow curve (i.e., r-Static-cost) correspond to the competitive ratios of the deterministic dynamic pricing mechanism developed in [TYBLG23] and the static randomized pricing mechanism in [SJTB24], respectively. In this figure, we set $v_{\min} = 1$, $v_{\max} = 10$, and $f(i) = \frac{i^2}{59}$.

the competitive ratio of r-Dynamic-cost approaches $\alpha^*_{\mathcal{S}}(k)$ as k goes to infinity, meaning that r-Dynamic-cost is asymptotically optimal.

Exact optimality of r-Dynamic-cost when $k = 2$. For the small inventory case of $k = 2$, a tighter analysis shows that r-Dynamic-cost is $\alpha^*_{\mathcal{S}}(2)$-competitive using the same design of pricing functions in Theorem 2.2, where $\alpha^*_{\mathcal{S}}(2)$ is the lower bound obtained in Theorem 2.1 for $k = 2$. This indicates that r-Dynamic-cost is not just asymptotically optimal, but also optimal in the small inventory setting when $k = 2$. The corollary below formalizes this result.

Corollary 2.1. *Given $\mathcal{S} = \{v_{\min}, v_{\max}, f\}$ for the* `kSelection-cost` *problem with $k = 2$,* r-Dynamic-cost *is $\alpha^*_{\mathcal{S}}(2)$-competitive when $\phi_1 : [0,1] \to [L_1, U_1]$ and $\phi_2 : [0,1] \to [L_2, U_2]$ are designed as follows:*

- *If $\alpha^*_{\mathcal{S}}(2) \geq \frac{2v_{\min}-c_1-c_2}{v_{\min}-c_1}$, then:*

$$\phi_1(s) = \begin{cases} v_{\min} & s \in [0, \xi^*], \\ (v_{\min} - c_1) \cdot e^{(s-\xi^*)\cdot\alpha^*_{\mathcal{S}}(2)/2} + c_1 & s \in [\xi^*, 1], \end{cases}$$
$$\phi_2(s) = (L_2 - c_2) \cdot e^{s\cdot\alpha^*_{\mathcal{S}}(2)/2} + c_2 \qquad \forall s \in [0,1].$$

In this case, the price intervals and ξ^* are given by

$$L_1 = v_{\min}, U_1 = L_2 = (v_{\min} - c_1) \cdot e^{(1-\xi^*)\cdot\alpha^*_{\mathcal{S}}(2)/2} + c_1, U_2 = v_{\max},$$
$$\xi^* = \frac{1}{\alpha^*_{\mathcal{S}}(2)} \cdot \frac{(2v_{\min} - c_1 - c_2)}{v_{\min} - c_1}.$$

- *If $\alpha^*_{\mathcal{S}}(2) < \frac{2v_{\min}-c_1-c_2}{v_{\min}-c_1}$, then:*

$$\phi_1(s) = v_{\min}, \qquad \forall s \in [0,1],$$
$$\phi_2(s) = \begin{cases} v_{\min} & s \in [0, \xi^*], \\ (v_{\min} - c_2) \cdot e^{(s-\xi^*)\cdot\alpha^*_{\mathcal{S}}(2)/2} + c_2 & s \in [\xi^*, 1]. \end{cases}$$

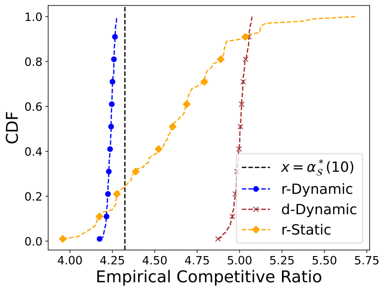
In this case, the price intervals and ξ^* are given by

$$L_1 = U_1 = L_2 = v_{\min}, \quad U_2 = v_{\max},$$
$$\xi^* = \frac{(2v_{\min} - c_1 - c_2)/\alpha^*_{\mathcal{S}}(2) - (v_{\min} - c_1)}{v_{\min} - c_2}.$$

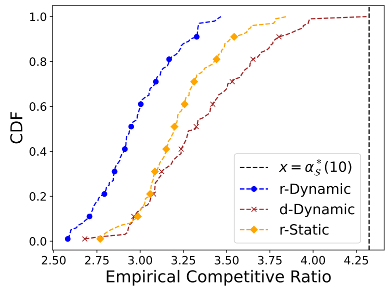
The proof of the corollary above is given in Appendix A.7. In the following two subsections, we first evaluate the empirical performance of r-Dynamic-cost and then provide a proof sketch of Theorem 2.2 to show the asymptotic optimality of r-Dynamic-cost.

2.3.2 Empirical Performance of r-Dynamic-cost

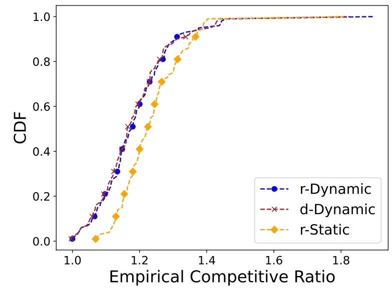
We perform three experiments to evaluate the empirical performance of r-Dynamic-cost and compare its performance to two other algorithms, d-Dynamic-cost [TYBLG23] and r-Static-cost [SJTB24]. Throughout the three experiments, the setup $\mathcal{S}$ is fixed to be $\{v_{\min} = 1, v_{\max} = 30, f(i) = i^2/16\}$ and $k = 10$. To stimulate different arrival patterns of buyers, we consider the



(a) **Instance-Sorted**



(b) **Instance-Low2High**



(c) **Instance-IID**

Figure 2.2: CDF plots of empirical competitive ratios of r-Dynamic-cost (Algorithm 1), d-Dynamic-cost [TYBLG23] and r-Static-cost [SJTB24].

following three types of instances:

- **Instance-IID**: We generate the valuations of 1000 buyers using the truncated normal distribution $N(15, 15)_{[1,30]}$.
- **Instance-Sorted**: We generate 1000 buyers using the same approach as **Instance-IID**, and sort these buyers in increasing order by their valuations. This instance mimics the hard instance $\mathcal{I}^{(\epsilon)}$.
- **Instance-Low2High**: We generate the valuations of 500 buyers using truncated normal distribution $N(7.5, 7.5)_{[1,30]}$. Following these 500 buyers, we generate another 500 buyers using distribution $N(22.5, 7.5)_{[1,30]}$.

Figure 2.2 presents the CDF plot of the empirical competitive ratios for the three algorithms r-Dynamic-cost, d-Dynamic-cost, and r-Static-cost, evaluated on 300 instances from each type of instance. In Figure 2.2(a), r-Dynamic-cost significantly outperforms the other two algorithms under **Instance-Sorted**. This is because the valuations of online arrivals are increasing, similar to the hard instance $\mathcal{I}^{(\epsilon)}$ defined in Section 2.2.2. This result confirms the superior performance of r-Dynamic-cost under difficult instances compared to the other algorithms. Additionally, Figure 2.2(a) demonstrates that r-Dynamic-cost's performance is very close to the lower bound $\alpha^*_{\mathcal{S}}(10)$, suggesting that r-Dynamic-cost may not only be asymptotically optimal in the large k regime but also near-optimal in the small k regime. In Figure 2.2(b), **Instance-Low2High** consists of two phases: low-valued buyers arriving first, followed by high-valued buyers. This instance is simpler than **Instance-Sorted**, and the performance of all three algorithms improves, with r-Dynamic-cost continuing to outperform the others. Finally, in Figure 2.2(c), under **Instance-IID**, all algorithms achieve a competitive ratio close to 1, with r-Dynamic-cost and d-Dynamic-cost performing similarly. These results indicate that r-Dynamic-cost's advantage is most evident on more challenging instances, particularly when low-valued buyers arrive before high-valued ones.

2.4 Chapter Summary

In this chapter, we address the Online Selection with Diseconomies of Scale (`kSelection-cost`), a setting where the marginal production costs increase as more units are produced and allocated. A key contribution is the derivation of a new, tight lower bound on the competitive ratio achievable by any online algorithm, explicitly characterizing the complexity introduced by diseconomies of scale. Building on this theoretical insight, we propose a randomized dynamic pricing algorithm, termed r-Dynamic-cost, that sequentially updates resource prices based on marginal production costs and remaining inventory. We analytically demonstrate that r-Dynamic-cost achieves optimal performance for small inventories and significantly outperforms existing algorithms in large-inventory scenarios. Empirical evaluations further support its robust performance, showing clear advantages over prior deterministic and static randomized methods. Whether Algorithm 1, introduced in Chapter 2, is optimal for all $k \geq 3$ remains an open question. While we conjecture that our randomized pricing mechanism achieves optimality for all $k \geq 1$, confirming its optimality for $k \geq 3$ demands a more refined and detailed analysis.

Chapter 3

Online k-Rental with Fixed Rental Duration

In this chapter, we study the online k-rental problem with fixed rental durations (`kRental-fixed`), where unlike the `kSelection-cost` problem studied in last chapter resource units are reusable. To address this problem, we propose a novel algorithm, rDynamic-fixed, which employs a randomized dynamic pricing strategy to generate fractional allocations that are subsequently rounded to integral allocations using our novel lossless online correlation rounding procedure (1-(k, d)-oca). We establish theoretical performance guarantees for rDynamic-fixed and derive a tight lower bound on the achievable competitive ratio, proving that rDynamic-fixed attains optimal competitiveness among all online algorithms for `kRental-fixed` problem.

3.1 Problem Formulation and Assumptions

In this section, we formally define the online k-rental with fixed rental duration (`kRental-fixed`) problem as follows. A decision maker has k units of a resource to allocate to online arriving buyers. The n-th online buyer, $n \in [N]$, arriving at time a_n, makes a request to rent one unit of the resource for a fixed rental duration d. If a unit of the resource is allocated to this buyer, the unit is rented for d time steps, and becomes available at time $a_n + d$ to be reused. The buyer has a valuation v_n for this request which is reported upon arrival, and the algorithm obtains value v_n if it accepts buyer n's request by allocating a unit of the resource to this buyer.

Let $x_n \in \{0, 1\}$ denote the decision to whether accept or reject buyer n's request by the algorithm. Then, the objective is to maximize the sum of the valuations of buyers to whom a unit of the resource is allocated, $\sum_{n\in[N]} x_n \cdot v_n$. This objective is to be maximized subject to the constraint

$$\sum_{j\in[n]} x_j \cdot 1_{\{d+a_j>a_n\}} \leq k, \quad \forall\, n \in [N].$$

The above constraint ensures that at any point in time throughout the horizon no more than k units of the resource are allocated. It is sufficient to satisfy this constraint at the arrival of each buyer n, when the available inventory may decrease.

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

$$\max_{\mathbf{x}} \quad \sum_{n\in[N]} v_n \cdot x_n, \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j\in[n]} x_j \cdot 1_{\{d+a_j>a_n\}} \leq k, \quad \forall\, n \in [N], \tag{3.2}$$

$$x_n \in \{0,1\}, \quad \forall\, n \in [N]. \tag{3.3}$$

At the arrival of each buyer, the algorithm must make an irrevocable decision to accept or reject the buyer's request. The uncertainty regarding future buyers' valuations and the overall demand for the resource makes this decision challenging. To achieve a bounded competitive ratio, we constrain the adversary by making the following assumption:

Assumption 3.1. *The buyers' valuations are within the range* $[v_{\min}, v_{\max}]$*,*
i.e., we have $v_n \in [v_{\min}, v_{\max}], \forall n \in [N]$*.*

Let $\mathcal{I}$ denote the set of all instances of the `kRental-fixed` problem that can be selected by an adversary while satisfying the above assumptions. Our goal is to design online algorithms whose objective value is competitive with that of `OPT`(I) for every instance $I \in \mathcal{I}$. The α-competitiveness of an online algorithm is defined according to Eq. (1.1). In the following section, we present the algorithm rDynamic-fixed for the `kRental-fixed` problem obtaining tight competitive ratio.

3.2 rDynamic-fixed: A Randomized Dynamic Pricing Algorithm for `kRental-fixed`

In this section, we introduce a randomized dynamic pricing algorithm (rDynamic-fixed) that utilizes 1-(k, d)-oca correlation scheme to obtain tight performance guarantees for the `kRental-fixed`.

Algorithm 2: Randomized Dynamic Pricing Algorithm with Lossless Online Correlation Procedure (rDynamic-fixed)

Input: Pricing function $\phi : [0,1] \to [v_{\min}, v_{\max}]$.
Initiate instance of 1-(k,d)-OCA procedure
for *each arriving buyer n* **do**
  Compute the expected utilization level of the inventory:

$$y_n = \sum_{j=1}^{n-1} x_j \cdot \mathbf{1}_{\{a_j + d > a_n\}}, \tag{3.4}$$

  Observe buyer n's valuation v_n.
  if $y_n < k$ **then**
    Compute the fractional allocation:

$$x_n = \underset{x \in \left[0, \min\{1, k - y_n\}\right]}{\arg\max} \left\{ x \cdot v_n - k \int_{\eta = y_n/k}^{(y_n + x)/k} \phi(\eta)\, d\eta \right\}. \tag{3.5}$$

    Decide the integral allocation by 1-(k,d)-OCA.
  end
  else
    Reject buyer n request
  end
end

rDynamic-fixed employs two key techniques to handle the uncertainty regarding buyers' valuations in the online market. First, it uses a continuous pricing function which is a function of the utilized fraction of the total inventory at the arrival time of the buyer. Using this pricing function, the algorithm produces, for each arriving buyer n, a fractional solution $x_n \in [0,1]$ that depends on the buyer's valuation and the current price of the resource. In the second step, the algorithm employs an online correlation procedure called 1-(k,d)-OCA, which takes x_n as input and correlates the current allocation decision with previous decisions to round the fractional solution x_n, so that a unit is allocated with probability exactly equal to x_n.

To be more specific, at the arrival of buyer n, the algorithm first computes the value y_n as defined in Eq. (3.4), which represents the expected number of resource units currently under rental by previous buyers whose rental intervals overlap with that of buyer n. Using this expected utilization level, the algorithm then applies the pricing function ϕ as described in Eq. (3.5) to determine the fractional allocation x_n. The term

$$k \int_{y_n/k}^{(y_n+x)/k} \phi(\eta)\, d\eta$$

in Eq. (3.5) can be interpreted as the total price of allocating an x-fraction of one unit of the resource, based on the current utilization level y_n/k of the total inventory and the pricing function ϕ. In this context, the fractional allocation x_n represents the optimal fraction of a resource unit to allocate to buyer n, given their valuation v_n and the pricing rule defined by ϕ.

Then, in the second step, x_n is passed to an online rounding procedure named 1-(k,d)-OCA. This procedure generates an integral decision whether to accept the buyer's request and allocate a unit of the resource or to reject that buyer. This decision is made by correlating the current allocation decision with previous ones. As long as $x_n \leq \min\{1,\, k - y_n\}$, based on Theorem 3.1, 1-(k,d)-OCA procedure allocates a unit to buyer n with probability exactly x_n and rounds the fractional solution x_n to a discrete decision.

Remark 3.1. *The 1-(k,d)-OCA procedure operates as an online algorithm, with an instance of this algorithm initiated at the start of Algorithm 2. Over its horizon, as each buyer n arrives at time a_n, the 1-(k,d)-OCA procedure receives a probability value $x_n \in [0,1]$ as input at time a_n. Based on this input, it makes an integral decision—either rejecting buyer n or allocating one unit of the resource—in a way that manages the resource inventory throughout the online process. A random seed is used to make these decisions; this seed is fixed when Algorithm 2 initiates the 1-(k,d)-OCA instance and remains the same for all buyers.*

Before delving into discussing theoretical performance guarantees for rDynamic-fixed, we first describe the 1-(k,d)-OCA procedure as we believe that the (k,d)-OCA rounding scheme is applicable to other settings and will be of independent interest.

3.3 Rounding Fractional Solutions to Integral Solutions: Online Correlated (k,d)-Assignment

To better explain the online rounding subroutine in line 8 of Algorithm 2, we formally define the Online Correlated (k,d)-Assignment scheme, dubbed (k,d)-OCA, and explain what a lossless online rounding procedure is. We then present the rounding procedure (k,d)-OCA in detail.

3.3.1 γ-(k,d)-OCA: Definitions, Objectives, and Challenges

Definition 3.1 (γ-(k,d)-OCA)**.** *Consider a set of k identical balls, each labeled uniquely from the set $\{1,2,\ldots,k\}$, where each ball can be assigned to a player for a fixed duration of d time units (after which it becomes available for reassignment). A sequence of N players arrives one at a time, with each player n characterized by a tuple (x_n,a_n), where $x_n \in [0,1]$ represents the target probability with which the procedure should assign a ball to player n, and a_n denotes the arrival time of player n. For any $\gamma \in [0,1]$, a γ-(k,d)-OCA is an online algorithm that, given the input sequence $\{(x_n,a_n)\}_{n\in[N]}$, assigns a ball to each player $n \in [N]$ with probability at least γx_n.*

By the definition above, γ-(k,d)-OCA is a randomized online algorithm that makes irrevocable decisions upon the arrival of each player, either allocating a ball to that player or rejecting it. The objective is to design γ-(k,d)-OCA for which the parameter γ is maximized.

Definition 3.2 (Lossless Rounding: 1-(k,d)-OCA)**.** *Given $k \geq 1$ and $d > 0$ with a sequence of players and input instance $\{(x_n,a_n)\}_{n\in[N]}$ satisfying Eq. (3.6) , a 1-(k,d)-OCA assigns a ball to player $n \in [N]$ with probability at least x_n.*

Based on the above, a 1-(k,d)-OCA is considered lossless if, unlike other correlation procedures that may be limited to allocating with probability only up to γx_n for some $\gamma < 1$, it guarantees allocation of a ball with the target probability x_n to each player n (i.e., $\gamma = 1$). For a given input instance of γ-(k,d)-OCA, $\{(x_n,a_n)\}_{n\in[N]}$, a necessary condition for the existence of a 1-(k,d)-OCA correlation procedure is the following constraint on the instance:

$$x_n \leq \min\left(1,\, k - \sum_{j\in[n-1]} x_j \cdot \mathbf{1}_{\{a_j+d>a_n\}}\right), \qquad \forall n \in [N]. \tag{3.6}$$

A formal statement of the above necessary condition will be given in Theorem 3.1. Here, we discuss some high-level intuitions. Note that $\sum_{j\in[n-1]} x_j \cdot \mathbf{1}_{\{a_j+d>a_n\}}$ represents the sum of targeted allocation probabilities of the players who arrived before player n and whose time interval (of length d), during which a ball is under their allocation if assigned, overlaps with that of player n. Moreover, the condition in Eq. (3.6) also ensures that $x_n \leq 1$ for all $n \in [N]$, so that no more than one ball is targeted for allocation to any individual player. If this constraint is violated, then a valid 1-(k,d)-OCA procedure cannot exist for such an instance, as the total targeted allocation probabilities at the arrival of player n would exceed the limited inventory of k balls available to the decision maker.

As seen above, one of the core challenges in designing a lossless γ-(k,d)-OCA (i.e., $\gamma = 1$) is to correlate the allocation decision for each newly arriving player with decisions previously made for players who still hold allocated resources. Such correlations must be managed carefully because resources currently rented by players become available at different future times. In the subsequent section, we propose a lossless online rounding scheme (Algorithm 3), whose central idea is to ***correlate each current allocation decision with past decisions involving players whose allocated resources return to the system sooner than others***.

3.3.2 A 1-(k, d)-OCA for `kRental-fixed`

In this subsection, we propose an online rounding scheme in Algorithm 3 and show that it is a 1-(k, d)-OCA if the sequence of targeted probabilities for each player satisfy the feasibility condition in Eq. (3.6).

Algorithm 3: Online Correlated 1-(k, d)-Assignment for `kRental-fixed`

Input: a sequence of fractional assignments $\{a_n, x_n\}_{n\in[N]}$.
Output: Integral assignment for each player $n \in [N]$.
Initialize: Set $m_1 = 1$, $p_1 = 0$; sample a random seed $r \sim \mathcal{U}(0, 1)$.
for *each player n arriving at time a_n* **do**
  Observe the probability value x_n.
  if $\sum_{j\in[n]} x_j \cdot \mathbf{1}_{\{a_j+d>a_n\}} > k$ **then**
    Reject player n.
  end
  if $p_n + x_n < 1$ **then**
    if $r \in [p_n, p_n + x_n)$ **then**
      Assign ball m_n to player n.
    end
    else
      Reject player n.
    end
    Update: $m_{n+1} = m_n$; $p_{n+1} = p_n + x_n$.
  end
  else
    if $r \in [p_n, 1]$ **then**
      Assign ball m_n to player n.
    end
    else if $r \in [0, x_n + p_n - 1)$ **then**
      Assign ball $m_n + 1$ to player n.
    end
    else
      Reject player n.
    end
    Update: $m_{n+1} = m_n + 1$ (if $m_{n+1} > k$, then set $m_{n+1} = 1$); $p_{n+1} = x_n + p_n - 1$.
  end
end

How Algorithm 3 Works: An Overview. The algorithm initially, before the arrival of any player, sets the random seed $r \sim \mathcal{U}(0, 1)$, which remains unchanged throughout the entire horizon of this online algorithm and is the only source of randomness in 1-(k, d)-OCA. Then, for all players, this random seed is used to decide whether to allocate a ball or not. Thus, this random seed helps create correlation among the decisions made for different players.

As each player n arrives, the algorithm maintains two pointers: $m_n \in \{1, 2, \ldots, k\}$, which indicates the next ball scheduled for allocation, and $p_n \in [0, 1]$, which determines a subrange of $[0, 1]$ for allocation decisions. Specifically, upon the arrival of player n, the algorithm acts according to the following two cases, depending on the values of p_n and x_n:

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

• **Case 2:** If $p_n + x_n \geq 1$, 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 $r \in [0, p_n + x_n - 1)$. Since $x_n \leq 1$, these intervals are non-overlapping, ensuring that no more than one ball is allocated.

Properties of Algorithm 3. We prove the following invariant regarding the availability of balls m_n and $m_n + 1$:

Proposition 3.1. *At the arrival of the n-th player in instance if the feasibility condition in Eq.* (**??**)*, the following holds:*

- *If $p_n + x_n < 1$, then if the random seed $r \in [p_n, p_n + x_n)$, ball m_n is available in the system.*
- *If $p_n + x_n \geq 1$, then:*
  - *If $r \in [p_n, 1]$, ball m_n is available in the system.*
  - *If $r \in [0, p_n + x_n - 1)$, ball m_{n+1} is available in the system (with the convention that if $m_n + 1 > k$, then $m_{n+1} = 1$).*

The proof of the above proposition can be found in Appendix B.3. It can be seen that this pointer mechanism synchronizes the scheduling of different balls for consecutive players. Furthermore, it keeps track of portions of $[0, 1]$ such that if r lies within those subranges, ball m_n or m_{n+1} remains available. Additionally, as explained for Algorithm 3, the total length of the subrange used for decision-making for each player n is exactly x_n. Since the random seed r is drawn from the uniform distribution $\mathcal{U}(0, 1)$ and a ball is allocated if r lies within these subranges, each player is allocated a ball with target probability x_n.

Theorem 3.1. *Given an instance $\{(x_n, a_n)\}_{n \in [N]}$, Algorithm 3 is a 1-$(k, d)$-OCA if the feasibility condition in Eq.* (3.6) *holds for each player n.*

We defer the full proof of the above theorem to Appendix B.4.

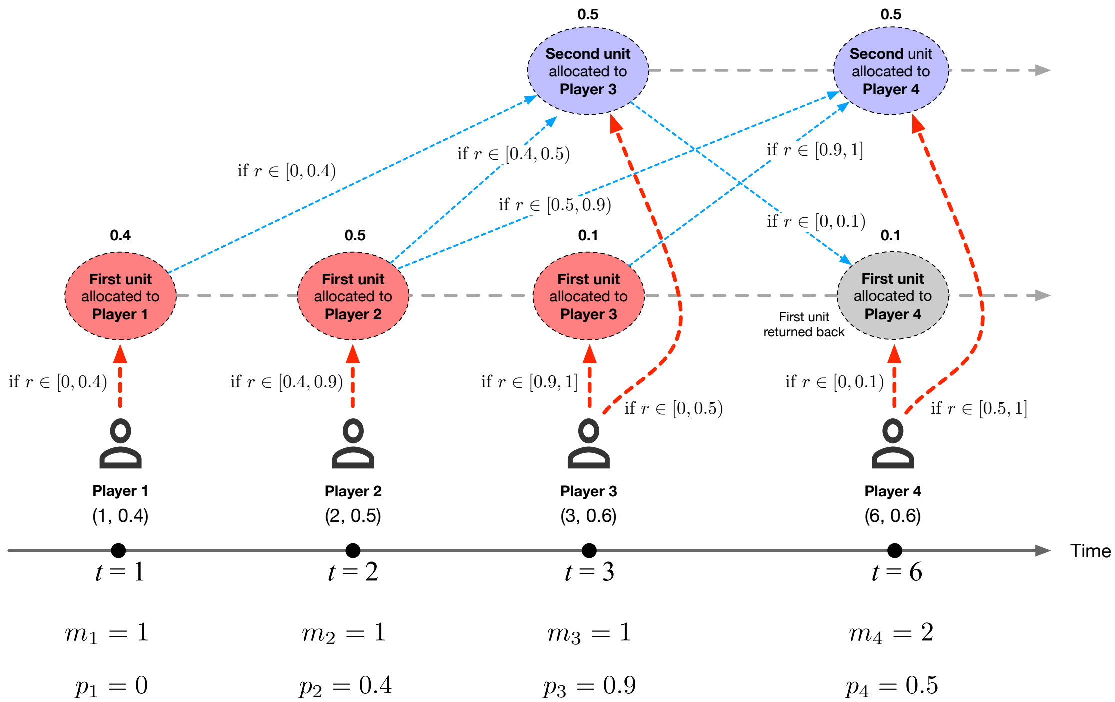


Figure 3.1: Illustration of the online correlated assignment process in Algorithm 3. The number above each circle represents the probability of the corresponding event occurring. In this example, player 3 receives the first ball with probability 0.1 and the second ball with probability 0.5. Notably, 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.

An Illustrative Example. To illustrate how Algorithm 3 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:

$$\Big\{(a_1 = 1, x_1 = 0.4), \quad (a_2 = 2, x_2 = 0.5), \quad (a_3 = 3, x_3 = 0.6), \quad (a_4 = 6, x_4 = 0.6)\Big\}.$$

It can be verified that for each player $n \in \{1, \cdots, 4\}$, the inequality in Eq. (3.6) 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 $x_1 = 0.4$, Algorithm 3 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 $x_2 = 0.5$, Algorithm 3 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 $x_3 = 0.6$, Algorithm 3 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 3 assigns the second ball if $r \in [0,\ 0.5)$. Consequently, **the total probability of allocating one ball to player 3 is 0.6**.

- Finally, when player 4 arrives at time $t = 6$ with $x_4 = 0.6$, Algorithm 3 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 3 always maintains a 1-(k, d)-OCA scheme in that it assigns one ball to every player $n \in [N]$ with probability exactly x_n.

3.4 Theoretical Guarantee of rDynamic-fixed

In this section, we present the theoretical upper bounds for rDynamic-fixed competitiveness over all instances of the `kRental-fixed` problem.

As explained in detail in Section 3.2, rDynamic-fixed employs the 1-(k, d)-OCA procedure to round the fractional allocation x_n at each time step and to make an integral decision. For this procedure to work correctly, the instance of the (k, d)-OCA problem that rDynamic-fixed provides as input must satisfy the feasibility conditions in Eq. (3.6).

Lemma 3.1. *Let $\{(a_n, x_n)\}_{n \in [N]}$ denote an input instance for 1-(k, d)-OCA generated by rDynamic-fixed. Then, the feasibility conditions in Eq. (3.6) holds for each player n in this instance.*

The above lemma naturally follows from the design of rDynamic-fixed. Using an online primal-dual framework, we prove the following performance guarantee regarding rDynamic-fixed.

Theorem 3.2. *rDynamic-fixed is $1 + \ln(\frac{v_{\max}}{v_{\min}})$-competitive for the `kRental-fixed` problem when the pricing function $\phi : [0, 1] \to [v_{\min}, v_{\max}]$ is given by*

$$\phi(y) = v_{\min} \cdot \exp\Big(\big(1 + \ln(\tfrac{v_{\max}}{v_{\min}})\big) \cdot y - 1\Big), \quad y \in [0, 1].$$

The proof of the above theorem can be found in Appendix B.1. The design of the pricing function ϕ follows from attempting to satisfy the dual constraint in a primal-dual framework used to prove the performance guarantee of rDynamic-fixed.

Next, we prove following proposition regarding the lower bound on the competitiveness of every online algorithm for the `kRental-fixed` problem, showing that rDynamic-fixed obtains optimal competitive ratio among all online algorithms.

Proposition 3.2. *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 provided in Appendix B.2. To establish this lower bound, we construct a family of hard instances analogous to those used for the `kSelection-cost` problem. Specifically, k buyers with valuation $v_{\min}$ arrive first, followed by additional batches of buyers whose valuations increase continuously up to $v_{\max}$. Furthermore, these buyers arrive within a short time interval relative to d, ensuring that any resource unit allocated to one buyer cannot be reallocated to later arrivals in these instances. By applying the representative function approach to this family of instances, we determine the optimal online algorithm for these set of hard instances, thereby deriving a lower bound on the performance of all online algorithms.

3.5 Chapter Summary

This chapter examines the `kRental-fixed` problem, in which a seller manages an inventory of k identical units of a resource available for rental to buyers arriving sequentially. Each buyer requests a resource for a fixed duration and has a distinct valuation for the request. The seller must immediately and irrevocably decide whether to accept each request without knowledge of future valuations, and we study this problem under the assumption of bounded valuations. This novel problem, introduced in this thesis, is motivated by the work in [DFNS22].

In this chapter, we present the novel randomized dynamic pricing algorithm, rDynamic-fixed, which generates fractional allocations using a pricing function. These fractional allocations are then transformed into integral decisions using a lossless online correlation assignment procedure (1-(k, d)-oca). This rounding procedure efficiently converts the fractional solutions into integral decisions, ensuring that rDynamic-fixed achieves the optimal competitive ratio. We provide a rigorous analysis of the algorithm's competitive ratio by establishing a matching lower bound and proving that, since rDynamic-fixed attains this bound, it is optimal.

Chapter 4

Online k-Rental with Variable Rental Durations

This chapter focuses on the `kRental-variable` problem, where buyers arrive online and request variable-length rentals of a limited, reusable resource, with valuations linear in rental duration. This setting generalizes the `kRental-fixed` problem studied in Chapter 3, capturing real-world scenarios such as cloud computing and equipment rental, where reusability and uncertain demand introduce significant challenges. We present the rDynamic-variable algorithm, a randomized dynamic pricing strategy that generates fractional allocations, which are then rounded online using a new limited-correlation scheme. Leveraging the LP-free certificate method, we establish strong competitive guarantees and prove the algorithm's order-optimality by deriving a lower bound on the performance of any online algorithm for the `kRental-variable` problem.

4.1 Problem Statement and Assumptions

We formally define online k-rental problem with variable rental durations (`kRental-variable`) as follows. In this problem, a decision-maker has k identical units of a resource to allocate to a sequence of N buyers who arrive online. The n-th buyer arrives at time a_n and requests one unit of the resource for d_n time steps. Thus, the requested rental duration d_n is buyer dependent. If a unit is allocated to buyer n, it remains occupied until time $a_n + d_n$, after which it becomes available for future requests. Each buyer's valuation is d_n, which is linear in the rental duration, and the algorithm obtains a reward of d_n by accepting buyer n's rental request.

Let x_n indicate whether buyer n rental request is accepted: $x_n = 1$ if accepted, and $x_n = 0$ otherwise. The objective is to maximize $\sum_{n=1}^{N} x_n \, d_n$, which is the sum of the valuations of the buyers whose requests are accepted. Additionally, the algorithm must satisfy the resource constraint

$$\sum_{j=1}^{n} x_j \mathbf{1}_{\{\, a_j + d_j > a_n\}} \le k, \quad \forall n \in [N].$$

In other words, the total number of buyers who arrived before buyer n, including buyer n, and still have a unit of the resource under their rental at time a_n, cannot exceed k. This constraint must hold whenever a new buyer arrives and a decision is made regarding the inventory of the resource and the state of the inventory changes, ensuring that no more than k resource units are allocated at any point in time.

Let $I = \{a_n, d_n\}_{n\in[N]}$ denote an instance of the problem, then, the performance of the optimal clairvoyant algorithm on instance I of `kRental-variable` problem, $\texttt{OPT}(I)$, can be computed based on following integer linear program:

$$\max_{\mathbf{x}} \quad \sum_{n\in[N]} d_n \cdot x_n, \tag{4.1}$$

$$s.t. \quad \sum_{j\in[n]} x_j \cdot 1_{\{d_j + a_j > a_n\}} \le k, \quad \forall n \in [N], \tag{4.2}$$

$$x_n \in \{0, 1\}, \quad \forall\, n \in [N]. \tag{4.3}$$

Every online algorithm, upon the arrival of each buyer, must make an irrevocable decision. The uncertainty regarding the rental duration of future buyers makes decision-making challenging in this problem. To achieve a bounded competitive ratio, we impose constraints on the adversary, ensuring that the requested rental durations of buyers in each instance of the problem selected by the adversary are bounded within a finite support.

Assumption 4.1. *The buyers' requested rental durations are within the range* $[d_{\min}, d_{\max}]$.

Let $\mathcal{I}$ denote the set of all instances of the problem `kRental-variable` that can be selected by an adversary while satisfying the above assumption. Our goal is to design online algorithms whose objective value is competitive with that of $\texttt{OPT}(I)$ for every instance $I \in \mathcal{I}$. The α-competitiveness of an online algorithm is defined according to Eq.(1.1). In the following section, we present the algorithm RDYNAMIC-VARIABLE for the `kRental-variable` problem obtaining order-optimal competitive ratio.

4.2 rDynamic-variable: A Randomized Algorithm with Limited Correlation

The challenging aspect of designing an online algorithm occurs in instances where, initially, a number of buyers arrive in a short period with rental durations close to $d_{\min}$ (the first wave of buyers). In this situation, the algorithm must allocate only a portion of its inventory to these early buyers due to the uncertainty regarding future demand for the resource. After some time, while the resources allocated to the first wave of buyers are still under rental, a second wave of buyers arrives with requests for longer rental durations close to $d_{\max}$. Since the algorithm obtains higher value by allocating its inventory to these later, higher-value buyers, it faces a challenging trade-off.

In the following, we present rDynamic-variable in Algorithm 4 for the `kRental-variable` problem. The general scheme of rDynamic-variable is as follows. At the arrival of each buyer, the algorithm uses the pricing function ϕ to determine the fractional allocation amount x_n of unit i_n^* to be allcated to buyer n. Then, the algorithm correlates the decision to allocate unit i_n^* to buyer n with previous decisions made regarding the allocation of this unit to earlier buyers, ensuring that, with probability exactly x_n, unit i_n^* is allocated to buyer n. In the following, we describe in detail how the fractional allocation x_n and the unit i_n^* are determined, and how the algorithm ensures that the unit i_n^* is allocated with probability exactly equal to x_n.

Step-I (Relax): Generating fractional allocation x_n. rDynamic-variable treats each unit of the resource uniquely. Upon the arrival of buyer n, the algorithm computes, for each unit i, the value $y_n^{(i)}$ according to Eq. (4.4), which is the probability that unit i is under the allocation of a previous buyer and is not available at the arrival of buyer n. We refer to $y_n^{(i)}$ as the *probabilistic utilization level* of unit i. Then, the algorithm determines unit i_n^* that has the lowest probabilistic utilization among all units at the arrival of buyer n. The algorithm computes a fractional value of $x_n \in [0, 1]$ according to Eq. (4.5). We can interpret the term

$$\int_{y_n^{(i_n^*)}}^{y_n^{(i_n^*)}+x} \phi(\eta)\, d\eta$$

as the price of an x-fraction of unit i_n^* according to the pricing function ϕ, given the current probabilistic utilization level $y_n^{(i_n^*)}$. Thus, the fractional allocation x_n can be interpreted as the maximum fraction of unit i_n^* (with utilization $y_n^{(i_n^*)}$) that is worth allocating to buyer n's request, which has a value d_n according to the pricing function ϕ. Thus, as described above, the pricing function ϕ is an increasing function of the probabilistic utilization of each unit. The function ϕ increases the price of a unit as a higher fraction of it is allocated, so that a larger portion of the unit is preserved for requests with longer rental durations and, hence, larger valuations.

Algorithm 4: Randomized Dynamic Pricing Algorithm with Limited Correlation for `kRental-variable` problem (rDynamic-variable)

Input: Pricing function ϕ

Initialization: Set the deterministic availability value $a_1^{(i)} = 1$ for each unit $i \in [k]$; set the probabilistic utilization level $y_1^{(i)} = 0$ for each unit $i \in [k]$.

while *a new buyer n arrives* **do**

  Observe the buyer's duration request, d_n.

  if *unit $i \in [k]$ has just been returned* **then**

    Set $a_n^{(i)} = 1$.

  end

  if $n > 1$ **then**

    For each unit $i \in [k]$ of the resource, compute probabilistic utilization level

$$y_n^{(i)} = \sum_{j=1}^{n-1} x_j \cdot \mathbf{1}_{\{a_j + d_j > a_n\}} \cdot \mathbf{1}_{\{i_j^* = i\}}. \quad (4.4)$$

  end

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

$$x_n = \arg\max_{x \in [0,1]} \Big\{ d_n \cdot x - \int_{y_n^{(i_n^*)}}^{y_n^{(i_n^*)} + x} \phi(\eta) d\eta \Big\}. \quad (4.5)$$

  if $a_n^{(i_n^*)} = 1$ **then**

    Allocate unit i_n^* to buyer n with probability $\frac{x_n}{1 - y_n^{(i_n^*)}}$.

    if *unit i_n^* successfully allocated to buyer n* **then**

      Set $a_n^{(i_n^*)} = 0$.

    end

  end

  Set $a_{n+1}^{(i)} = a_n^{(i)}$ for each unit $i \in [k]$.

end

Step-II (Round): Rounding fractional solutions $\{x_n\}_{n\in[N]}$ to integral solutions. In the second phase, once the fractional allocation x_n is determined, the algorithm correlates the current allocation decision for unit i_n^* with its past decisions. Specifically, when unit i_n^* is available, the algorithm employs a new random seed to allocate it to buyer n with probability $\frac{x_n}{1-y_n^{(i_n^*)}}$, thereby ensuring that the unit is allocated to buyer n with probability exactly x_n. This correlation process treats each resource unit independently, creating correlation only among the decisions associated with that particular unit. add a note here that the correlation scheme in above is integrated within the design of pricing function

Remark 4.1. *It is important to note that the correlation scheme of the 1-(k, d)-OCA procedure in Algorithm 3 cannot be extended to settings with buyer-dependent rental durations; indeed, when the duration of the requested rental intervals varies (i.e., is request-dependent), no lossless 1-(k, d)-OCA scheme exists. Consequently, we develop a correlation scheme for rDYNAMIC-VARIABLE that employs a more limited correlation procedure compared to 1-(k, d)-OCA. In rDYNAMIC-VARIABLE, the decision for each buyer is correlated only with a limited subset of previous buyers—specifically, those who received a fractional allocation of unit i_n^*—whereas in Algorithm 2 the decisions for all buyers are correlated and synchronized.*

4.3 Theoretical Guarantees of rDYNAMIC-VARIABLE

In the following proposition, the constraints that the design of the ϕ function needs to satisfy to obtain α-competitiveness for the `kRental-variable` problem are discussed:

Proposition 4.1. *rDYNAMIC-VARIABLE is α-competitive for the `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 $n \in [N]$:*

$$\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\,\phi(\eta)\,d\eta \;+\; \frac{\alpha}{3}\,d_n\left(\phi^*(d_n) \;-\; 2\,y_1\right) \;\geq\; d_n, \quad \forall\, y_1 \in \left[0,\, \tfrac{\phi^*(d_n)}{2}\right], \tag{4.6}$$

$$\frac{\alpha}{3}\,d_n\,\phi^*(d_n) \;-\; \frac{\alpha}{3}\,y_2\left(d_n \;-\; \phi(y_2)\right) \;\geq\; d_n, \quad \forall\, y_2 \in \left[0,\, \phi^*(d_n)\right]. \tag{4.7}$$

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

By Proposition 4.1, the design of an algorithm with strong performance essentially reduces to finding a pricing function ϕ that satisfies the constraints in Eqs. (4.6) and (4.7) while minimizing α. The above constraints follow from the dynamics of the problem and the assumption that buyers can arrive at any point in time throughout the horizon. The detailed proof of the above proposition can be found in Appendix C.1.

To design a pricing function that satisfies the constraints in Proposition 4.1 with the lowest possible value of α, it is sufficient to enforce these constraints only at the critical points where the left-hand side of the inequalities is minimized. Moreover, it can be shown that for the pricing function to achieve the minimum value of α, the inequality must hold with equality. This leads us to a system of differential equations that include delayed terms. In particular, the fact that the pricing function obtaining tightest competitive ratio based on above constraints is derived from a delayed differential equation indicates that past prices, which reserve a fractional of the resource for low-duration buyers, are used to influence current price dynamics. This delay introduces memory effects, making the pricing function sensitive to the duration of accepted requests that have been allocated a fractional portion of the inventory. In effect, the pricing function not only determines the current price but also manages the scheduling of buyers' requests by balancing resource reservations. This balance takes into account the current inventory that is allocated to buyers with short rental durations, who are likely to return the resources soon.

In the following corollary, we present an analytical design of ϕ that satisfies the system of constraints in Proposition 4.1.

Corollary 4.1. *rDynamic-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:*

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

The proof of the above corollary can be found in Appendix C.3. Corollary 4.1 follows from Proposition 4.1 since the design of the ϕ function presented above satisfies the constraints in Eq. (4.6) and Eq. (4.7). However, this design is not the optimal one, as it does not minimize α. Obtaining a closed-form solution for ϕ that minimizes α while satisfying these constraints is challenging because it involves solving a system of delayed differential inequalities with an inverse term. Therefore, we resort to numerical methods to construct such a pricing function ϕ.

To achieve this, 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 Proposition 4.1 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 $\{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}$.

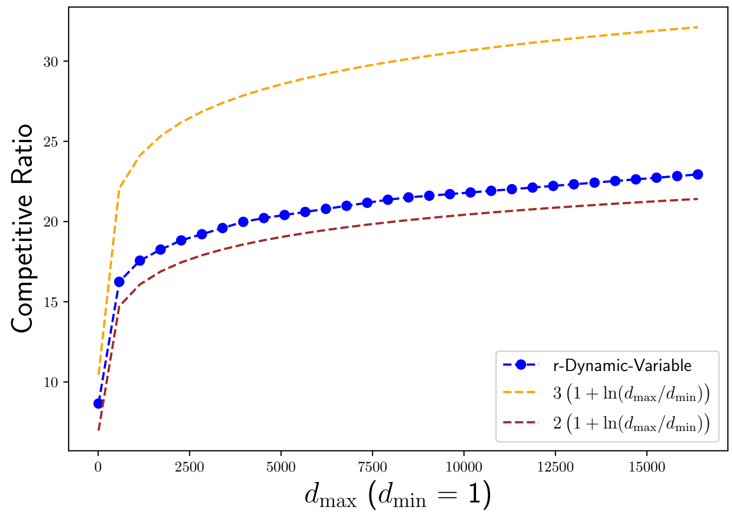


Figure 4.1: The competitive ratio of Algorithm 4 as $d_{\max}$ increases, for the pricing function ϕ designed according to Proposition 4.2, with $\epsilon = 0.001$.

Proposition 4.2. *For any given $\epsilon \in (0,1)$, rDynamic-variable is α^*-competitive for the* `kRental-variable` *problem if the pricing function ϕ is designed as*

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

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:

$$\alpha^* = \min_{\alpha,\ \{\pi_i^{(\epsilon)}\}_{i=1}^{\lceil 1/\epsilon \rceil}} \alpha \tag{4.8a}$$

$$s.t. \quad \pi_i^{(\epsilon)} \le \pi_{i+1}^{(\epsilon)}, \quad \forall i \in \left\{1, 2, \dots, \lceil \tfrac{1}{\epsilon} \rceil - 1\right\}, \tag{4.8b}$$

$$\pi_1^{(\epsilon)} = d_{\min}, \tag{4.8c}$$

$$\pi^{(\epsilon)}_{\lceil \frac{1}{\epsilon} \rceil} \ge d_{\max}, \tag{4.8d}$$

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

$$\forall i \in \left\{1, \dots, \lfloor \tfrac{l}{3} \rfloor\right\},\ l \in \left\{\left\lceil \tfrac{3}{\alpha\epsilon} \right\rceil, \left\lceil \tfrac{3}{\alpha\epsilon} \right\rceil + 1, \dots, \left\lceil \frac{1}{\epsilon} \right\rceil\right\}, \tag{4.8e}$$

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

$$\forall i \in \{1, \dots, l\},\ l \in \left\{\left\lceil \tfrac{2}{\alpha\epsilon} \right\rceil, \left\lceil \tfrac{2}{\alpha\epsilon} \right\rceil + 1, \dots, \left\lceil \frac{1}{\epsilon} \right\rceil\right\}. \tag{4.8f}$$

The proof of the above proposition is given in Appendix C.2. The above optimization problem can be solved by standard LP solvers[1] for different values of ϵ, $d_{\min}$, and $d_{\max}$. For $\epsilon = 0.001$ and $d_{\min} = 1$, Figure 4.1 shows the competitive ratio—equal to α^*, the optimal value of the LP in Proposition 4.2—for rDynamic-variable when the ϕ function is chosen according to Proposition 4.2. As illustrated, numerical methods can produce a pricing function design that offers a stronger performance guarantee than the one proposed in Corollary 4.1.

We can prove the following lower-bound on the competitivness of any online algorithm for `kRental-variable` problem.

Proposition 4.3 (Lower Bound of `kRental-variable`). *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.*

The proof of the above proposition can be found in Appendix C.4. The idea is to design a set of hard instances in which initially k buyers with rental duration $d_{\min}$ arrive. Following these buyers, there will be additional batches of buyers whose rental durations increase continuously until reaching $d_{\max}$. Moreover, these buyers arrive in a very short span of time. Following from the above lower bound result and Corrolary 4.1, we argue that rDynamic-variable is order-optimal.

4.4 Chapter Summary

In this chapter, we study the `kRental-variable` problem, where a decision maker manages reusable resources rented to sequentially arriving buyers with buyer-dependent rental durations. We present a novel algorithm, rDynamic-variable, which employs a randomized dynamic pricing strategy to generate fractional solutions that address uncertainties in both valuations and rental durations. These fractional solutions are then efficiently rounded using our newly developed online rounding scheme. Through detailed theoretical analysis, we derive performance guarantees and prove that the algorithm's competitiveness is order-optimal. Our algorithm achieves tighter performance guarantees compared to those developed in [GL16, GG20] for a problem similar to `kRental-variable`, yielding a better competitive ratio in both small and large inventory settings. We derive pricing functions using an alternative approach for resource allocation that is memory-dependent and accounts for the pricing of the resource based on the rental durations of current buyers, who may receive a unit of the resource and return it at an early time.

However, it remains an open problem to develop algorithms with even tighter performance guarantees and to establish stronger lower bounds for any online algorithm solving the `kRental-variable`

[1] It is worth noting that the number of variables in above LP grows on the order of $\lceil 1/\epsilon \rceil$, while the number of constraints increases on the order of $(\lceil 1/\epsilon \rceil)^2$. Techniques such as the interior point method can be employed to solve this optimization problem efficiently.

problem. Furthermore, it is still unclear how to derive correlation schemes that can round fractional solutions more effectively than the limited-correlation approach used in Algorithm 4. An important open question is how to extend the online correlation assignment ((k, d)-OCA) procedure developed in Chapter 3 so that it can efficiently round fractional allocations in settings with variable, buyer-dependent rental durations.

Part II

ONLINE MATCHING

Chapter 5

Online Edge-Weighted Matching with Bounded Weights

This chapter addresses the online edge-weighted matching with bounded weights (`Matching-BW`) problem, focusing on settings where a decision-maker manages multiple resource types, each with a limited number of units, which must be allocated immediately and irrevocably to sequentially arriving buyers. Each buyer has distinct, bounded valuations for different resources that are revealed upon arrival. The chapter introduces a randomized static pricing algorithm, rSTATIC-MATCHING, which maintains fixed prices throughout the buyer arrival period to avoid price discrimination. Through rigorous theoretical analysis, we derive new competitive ratio guarantees and establish a tight lower bound, demonstrating that rSTATIC-MATCHING achieves optimal performance among all online algorithms using static pricing.

5.1 Problem Statement and Assumptions

Let us define the online edge weighted matching problem (`Matching-BW`) as follows. In this problem, a seller maintains J resources, with each resource j having an inventory of C_j identical units for sale. N buyers arrive sequentially. Upon the arrival of each buyer $n \in [N]$, the seller posts a price $p_{n,j}$ for each resource $j \in [J]$. Buyer n has private valuations over the resources $\{v_{n,j}\}_{j\in[J]}$, where $v_{n,j}$ represents her valuation for one unit of resource j. Without loss of generality, $v_{n,j} = 0$ if buyer n is not interested in resource j. Based on the posted prices, buyer n can obtain a utility of $v_{n,j} - p_{n,j}$ from purchasing resource j. Then, buyer n decides to purchase resource

$$j^* = \arg\max_{j\in[J]} \Big\{ v_{n,j} - p_{n,j} \Big\},$$

provided that the utility from purchasing resource j^* is non-negative.

Let $x_{n,j} \in \{0, 1\}$ denote the decision of whether buyer n purchases resource j (with $x_{n,j} = 1$ indicating a purchase and $x_{n,j} = 0$ otherwise). Buyer n obtains a utility of

$$u_n = \sum_{j \in [J]} x_{n,j}(v_{n,j} - p_{n,j}),$$

and the seller collects a total revenue of

$$r_j = \sum_{n \in [N]} x_{n,j} p_{n,j}$$

from selling resource j. The overall objective is to maximize the social welfare, sum of utilities of buyers and the seller, which is given by

$$\sum_{n \in [N]} u_n + \sum_{j \in [J]} r_j = \sum_{n \in [N]} \sum_{j \in [J]} x_{n,j} v_{n,j}.$$

We denote by $I := \{v_{n,j}\}_{n \in [N], j \in [J]}$ an instance of the `Matching-BW` problem. Given this instance, the optimal social welfare of a clairvoyant algorithm, $\texttt{OPT}(I)$, can be obtained by solving the following offline problem:

$$\max_{x_{n,j}} \quad \sum_{n \in [N]} \sum_{j \in [J]} v_{n,j} x_{n,j} \tag{5.1}$$

$$\text{s.t.} \quad \sum_{n \in [N]} x_{n,j} \le C_j, \quad \forall j \in [J], \tag{5.2}$$

$$\sum_{j \in [J]} x_{n,j} \le 1, \quad \forall n \in [N], \tag{5.3}$$

$$x_{n,j} \in \{0, 1\}, \quad \forall n \in [N],\ \forall j \in [J], \tag{5.4}$$

where constraint (5.2) guarantees that at most C_j copies of resource j are sold, and constraint (5.3) ensures that each buyer is allocated at most one unit from one of the resources.

In the online setting, a posted price algorithm must determine the price vector $\{p_{n,j}\}_{j \in [J]}$ for each buyer n just based on past purchase decisions $\{x_{i,j}\}_{j \in [J], i < n}$ without the information of future buyers. We aim to find the online pricing algorithm that can achieve the minimum competitive ratio.

It is well-known that no online algorithm can achieve a bounded competitive ratio in the general online edge-weighted matching problem [M+13]. Therefore, prior work relies on various additional assumptions based on specific applications. For example, this problem has been studied under the free disposal assumption in the Ad assignment problem [FKMP09]. In this paper, we extend the bounded valuation assumption of pervious chapters to `Matching-BW`.

Assumption 5.1. *Buyers' valuations are bounded if they are interested in resource* j*, i.e.,* $v_{n,j} \in [v_{\min,j}, v_{\max,j}], \forall n \in [N], j \in [J]$.

Let $\theta_j = v_{\max,j}/v_{\min,j}$ denote the fluctuation ratio of resource j, and let $\theta_{\max} = \max_{j\in[J]} \theta_j$. It is worth noting that the online vertex-weighted matching problem, first introduced by [AGKM11], is a special case of the `Matching-BW` problem in which, for each resource type j, $\theta_j = 1$.

5.2 rStatic-matching: Static Pricing Algorithm for `Matching-BW`

In the following, we present rStatic-matching, a posted-price mechanism that uses a randomized static pricing scheme. Initially, before the arrival of any buyer, the algorithm samples a set of random seeds $\{s_j\}_{j\in[J]}$, independently drawn from the uniform distribution $\mathcal{U}(0,1)$ for each resource j. Based on these random seeds, the algorithm sets the static prices $p_j = \phi_j(s_j)$ for each resource j, where ϕ_j denote the pricing function used to allocate resource type j. Throughout the horizon, these prices remain fixed and are posted to each arriving buyer.

When a buyer arrives with valuation vector $\{v_{n,j}\}_{j\in[J]}$, she determines the resource j^* that maximizes her utility

$$\max_{j\in[J]} \Big(v_{n,j} - p_j\Big).$$

She then purchases resource j^* if the resulting utility $v_{n,j^*} - p_{j^*}$ is nonnegative and rejects all prices otherwise. It is worth mentioning that buyer n may choose not to purchase resource j even if her valuation $v_{n,j}$ is higher than the posted price p_j of resource j. This is because rStatic-matching may instead select another resource that can provide greater utility. Such competition among resources leads to the correlation of the online decision for one resource and the posted prices of the other resources.

5.3 Theoretical Grantee of rStatic-matching

By carefully designing the set of pricing functions $\{\phi_j\}_{j\in[J]}$, rStatic-matching can attain the competitive ratio discussed in the following theorem.

Algorithm 5: Randomized Static Pricing Algorithm (rStatic-matching) for `Matching-BW`

Input: A set of pricing functions $\{\phi_j(\cdot)\}_{j\in[J]}$.
Initialization: For each resource $j \in [J]$: Sample a random seed $s_j \sim \mathcal{U}(0,1)$ independently; set the static price $p_j = \phi_j(s_j)$;
Initialize the inventory counter $y_j = 1$.
for *each arriving buyer* n*:* **do**
  Define the set of available resources: $A = \{j \in [J] \mid y_j \leq C_j\}$.
  Post the prices $\{p_j\}_{j\in A}$ for the resource types in A to buyer n.
  if *buyer* n *purchases a unit of resource type* $j^* \in A$ **then**
    Set $x_{n,j^*} \leftarrow 1$ and $x_{n,j} \leftarrow 0$ for all $j \neq j^*$.
    Update the inventory counter: $y_{j^*} \leftarrow y_{j^*} + 1$.
  end
  else
    Set $x_{n,j} \leftarrow 0$ for all $j \in [J]$.
  end
end

Theorem 5.1. *rStatic-matching is $\alpha^*_{\texttt{Matching-BW}}$-competitive for the `Matching-BW` problem when the set of pricing functions $\{\phi_j\}_{j\in[J]}$ is given by, for all $j \in [J]$,*

$$\phi_j(y) = \begin{cases} \dfrac{(\alpha_j - 1)v_{\min,j}}{\alpha_j} \cdot e^y & \text{if } y \in [0, \omega_j), \\ v_{\max,j} e^{\alpha_j(y-1)} & \text{if } y \in [\omega_j, 1], \end{cases} \tag{5.5}$$

where ω_j is the solution of

$$\frac{e^{\omega_j}}{e^{\omega_j} - 1} = \frac{\ln \theta_j}{1 - \omega_j}, \tag{5.6}$$

$\alpha_j = \dfrac{e^{\omega_j}}{e^{\omega_j} - 1}$*, and* $\alpha^*_{\texttt{Matching-BW}} = \max_{j\in[J]} \alpha_j$.

We extend the economics-based approach developed by [EFFS21] to analyze rStatic-matching. The proof details are given in Appendix D.1.

It should be noted that rStatic-matching obtains a competitive ratio of $\frac{e}{e-1}$ in the online vertex-weighted matching problem, which is a special case of the `Matching-BW` problem where for each resource type $\theta_j = 1$ (that is, $v_{\max,j} = v_{\min,j}$). In this case, based on Theorem 5.1, as θ_j converges to one, the corresponding ω_j, which is the solution to Eq. (5.6), also approaches one. Thus, the competitive ratio of Algorithm 5 also converges to $\frac{e}{e-1}$, which is the optimal competitive ratio that can be obtained by any online algorithm for the online vertex-weighted matching problem [AGKM11].

5.4 Lower Bound of `Matching-BW`

In this section, we establish a lower bound for `Matching-BW` that matches the competitive ratio $\alpha^*_{\texttt{Matching-BW}}$ achieved by rSTATIC-MATCHING, as stated in Theorem 5.1. Consequently, our proposed static pricing algorithm is optimal among all online algorithms.

Theorem 5.2. *No online algorithm, including deterministic and randomized algorithms, for* `Matching-BW` *problem can achieve a competitive ratio smaller than* $\frac{e^\omega}{e^\omega-1}$*, where* ω *is the solution of* $\frac{e^\omega}{e^\omega-1} = \frac{\ln\theta_{\max}}{1-\omega}$.

We still rely on a representative function approach to demonstrate the lower bound of `Matching-BW` similar to the proof done for obtaining a lower-bound for `kSelection-cost` problem in Chapter 2. We first construct a family of hard instances, where each instance can be divided into two stages. The instance in Stage I is the classic upper-triangle instance from the online matching literature [DJ12], which requires the online algorithm to balance the numbers of sold units from different resources. The Stage II of the instance follows the design of the worst-case instance for obtaining a lower-bound for `kSelection-cost` problem in Chapter 2, which requires the online algorithm to reserve some units for high-valuation buyers. Because of the possible occurrence of the Stage II, algorithm cannot fully utilize its inventory for each of the resources in the first stage. Let ω denote the maximum utilization reserved for resources in Stage I, the competitive ratio of any algorithms is lower bounded by $\alpha \geq \frac{e^\omega}{e^\omega-1}$, following similar arguments in the lower bound proof of online matching. At the end of Stage I, the arrivals of Stage II are constructed similarly to the worst-case instance for obtaining a lower-bound for `kSelection-cost` problem in Chapter 2, and only interested in one of the resources that have been sold up to ω. We modify the lower bound proof of `kSelection-cost` problem by additionally taking into account the selection decisions in Stage I, and lower bound the competitive ratio by $\alpha \geq \frac{\ln\theta}{1-\omega}$. The lower bound of `Matching-BW` is obtained by optimizing the representative function to balance the difficulties from the instances in the two stages. The proof details are presented in Appendix D.2.

5.5 Chapter Summary

In this chapter, we study the online edge-weighted matching with bounded weights (`Matching-BW`) problem, where a seller allocates limited units from multiple resource types to sequentially arriving buyers, each with distinct and bounded valuations. To address concerns regarding fairness and price discrimination, we introduce the randomized static pricing algorithm, rSTATIC-MATCHING, which maintains fixed yet randomized resource prices throughout the buyer arrival period. We rigorously analyze the competitive performance of this algorithm, deriving a tight lower bound on its competitive ratio, and demonstrate that rSTATIC-MATCHING achieves optimal competitiveness not only among static pricing mechanisms but also among all online algorithms designed for the

`Matching-BW` problem. This result underscores the feasibility of ensuring fairness without sacrificing allocation efficiency in adversarial online matching scenarios using randomized pricing schemes.

Chapter 6

Online Edge-Weighted Matching with Reusable Resources

In this chapter, we study the online edge-weighted matching with reusable resources (`Matching-RR`) problem, which generalizes the problem setting studied in Chapter 4 (`kRental-variable`) to multiple resource types, each with an inventory of multiple reusable units. Buyers arrive sequentially, requesting to rent resources for varying durations, each with distinct per-unit-time valuations. We propose a randomized hybrid pricing algorithm, rHybrid-matching-RR, combining static randomized pricing to handle uncertainties in valuations and dynamic pricing to manage variability in rental durations. We rigorously analyze the performance of this algorithm using a randomized LP-free certificate method, and then deriving a new lower bound to demonstrate the order-optimality of our algorithm.

6.1 Problem Statement and Assumptions

Let us formally define the online edge-weighted matching with reusable resources (`Matching-RR`). A seller manages J distinct resources, where resource type $j \in [J]$ has an inventory of C_j identical units. A sequence of N buyers arrives *online*, one at a time. Upon the arrival of buyer n at time a_n, the buyer reports a collection of resource-specific parameters $\left(d_{n,j}, \rho_{n,j}\right)_{j\in[J]}$, where $d_{n,j}$ is the duration for which buyer n wishes to rent one unit of resource j, and $\rho_{n,j}$ is buyer n's per-unit-time valuation for renting a unit of resource j. Hence, the algorithm obtains total value $\rho_{n,j} \cdot d_{n,j}$ by allocating one unit of resource j to buyer n. Whenever either $d_{n,j} = 0$ or $\rho_{n,j} = 0$, buyer n has no interest in resource j. If a unit of resource type j is allocated to buyer n, it remains under buyer n's rental for $d_{n,j}$ time steps and becomes available again at time $a_n + d_{n,j}$.

When buyer n arrives, the seller must make an irrevocable decision to decide whether to *reject* this buyer or to allocate *at most one* unit from a resource j for duration $d_{n,j}$. Following the convention in online matching, we assume that each buyer can at most rent one unit from one of the resources.

Let $x_{n,j} \in \{0,1\}$ denote the decision of the online algorithm to allocate a unit of resource j to buyer n. The total objective value of the online algorithm is equal to $\sum_{n=1}^{N} x_{n,j}\, \rho_{n,j}\, d_{n,j}$, which is the sum of the valuations of the buyers whose request is accepted. Furthermore, to guarantee that no more than C_j unit of each resource is allocated at any point of time throughout the horizon, the algorithm must satisfy the following constraint

$$\sum_{\tau=1}^{n} x_{\tau,j}\, \mathbf{1}_{\left\{a_\tau + d_{\tau,j} > a_n\right\}} \leq C_j, \forall n \in [N], \forall j \in [J].$$

Let $I = \{(d_{n,j}, \rho_{n,j})_{j\in[J]}, a_n\}_{n\in[N]}$ denote an instance of the `Matching-RR` problem. The optimal clairvoyant algorithm's performance on instance I is given by solving the following integer linear program:

$$\begin{aligned}
\max_{x_{n,j}} \quad & \sum_{n\in[N]} \sum_{j\in[J]} \rho_{n,j}\, d_{n,j}\, x_{n,j}, && (6.1a)\\
\text{s.t.} \quad & \sum_{\tau=1}^{n} x_{\tau,j}\, \mathbf{1}_{\left\{a_\tau + d_{\tau,j} > a_n\right\}} \leq C_j, \quad \forall\, n \in [N],\ \forall\, j \in [J], && (6.1b)\\
& \sum_{j\in[J]} x_{n,j} \leq 1, \quad \forall\, n \in [N], && (6.1c)\\
& x_{n,j} \in \{0,1\}, \quad \forall\, n \in [N], \forall j \in [J]. && (6.1d)
\end{aligned}$$

In `Matching-RR`, there is an additional layer of uncertainty compared to `kRental-variable`, as buyers have distinct per-unit-time valuations for each resource j. In `kRental-variable`, every buyer has a fixed per-unit-time valuation of one. However, in `Matching-RR`, each buyer may hold a different valuation for each resource type. Moreover, the seller's inventory now comprises multiple resource types, subject to a matching constraint that limits each arriving buyer to receiving at most one resource unit.

To ensure a bounded competitive ratio, we impose the following assumptions on `Matching-RR`, which must hold for each resource type $j \in [J]$:

Assumption 6.1 (Bounded Valuations)**.** *For all buyers n, the valuation per unit time for resource j lies within $[\rho_{\min,j}, \rho_{\max,j}]$.*

Assumption 6.2 (Bounded Durations)**.** *For all buyers n, the requested rental duration for resource j lies in the range $[d_{\min,j}, d_{\max,j}]$.*

Let $\mathcal{I}$ denote the set of all instances of the `Matching-RR` problem that can be selected by an adversary while satisfying the above assumption. The goal is to design online algorithms whose objective value is competitive with that of $\texttt{OPT}(I)$ for every instance $I \in \mathcal{I}$. The α-competitiveness of an online algorithm is defined according to Eq. (1.1).

Remark 6.1. *The objective in* `Matching-RR` *can be interpreted as finding a maximum edge-weighted matching in an online bipartite graph, where the offline nodes are reusable. Specifically, each resource type j is viewed as an offline node with C_j identical copies, and each buyer n is an online node arriving sequentially. The set of resource types for which the buyer's requested rental is nonzero forms the set of neighbors for the online node corresponding to that buyer. Moreover, the valuation $d_{n,j} \cdot \rho_{n,j}$ obtained from allocating one unit of resource j to buyer n is interpreted as the weight of the edge that matches the online node (representing buyer n) to the offline node (representing a unit of resource type j). The set of edges for each online node and their corresponding weights are revealed upon that node's arrival. Thus, the objective of maximizing the total valuation of accepted requests can be seen as finding the maximum edge-weighted matching in this bipartite graph, under the condition that each offline node is reusable once its allotted rental duration elapses.*

6.2 rHybrid-matching-RR: A Randomized Algorithm for `Matching-RR`

In this section, we present a randomized algorithm, dubbed rHybrid-matching-RR, in Algorithm 6 for the `Matching-RR` problem, which incorporates both static and dynamic pricing schemes. We first provide an explanation of the algorithm, and then in the following section, we establish theoretical guarantees on the algorithm's competitive ratio.

Two sources of uncertainty regarding buyers' rental durations and their per-unit-time valuations make the design of online algorithms in this problem challenging. Furthermore, compared to the `kRental-variable` problem, we also have different resource types and a matching constraint that each buyer can be allocated (matched) at most one unit; the uncertainty regarding the total demand with respect to each resource type adds an additional layer of uncertainty.

In Algorithm 6, we handle the uncertainty regarding the buyer's rental duration using the algorithm developed for the `kRental-variable` problem. Furthermore, to address the uncertainty with respect to the buyer's per-unit-time valuation, we use a randomized static price for each resource type that is set at the beginning, before the arrival of the first buyer, and we only accept a rental request for a resource type if the buyer's per-unit-time valuation for that resource type is at least equal to that static price. Finally, to handle the trade-off with respect to the demand for different resource types at the arrival of each buyer, we select a resource type for allocation based on the value obtained by the algorithm if a unit of that resource type is allocated, as well as the available fraction of inventory for that resource type.

Algorithm 6: Randomized Hybrid Static-Dynamic Pricing Algorithm for `Matching-RR` (rHybrid-matching-RR)

Input: Two sets of pricing functions $\{\phi_{1,j}\}_{j\in[J]}$ and $\{\phi_{2,j}\}_{j\in[J]}$.
Initialization:
Sample a random seed vector $\mathbf{s} = \{s_j\}_{j\in[J]}$, where $s_j \sim \mathcal{U}(0,1)$; for each resource $j \in [J]$, set $p_j = \phi_{2,j}(s_j)$; for each resource $j \in [J]$ and each unit $i \in [C_j]$ of resource j,
Set availability variable $a_j^{(i)} = 1$ for each $j \in [J]$ and $i \in [C_j]$.
Set utilization level variable $y_{1,j}^{(i)} = 0$ for each $j \in [J]$ and $i \in [C_j]$.
for *each arriving buyer n* **do**
  if *a previously allocated unit i of resource j is returned* **then**
    $a_j^{(i)} \leftarrow 1$.
  end
  Let $R = \{j \in [J] : \rho_{n,j} \geq p_j\}$.
  for *each resource $j \in [R]$ and each unit $i \in [C_j]$* **do**

$$y_{n,j}^{(i)} = \sum_{\kappa=1}^{n-1} x_\kappa \cdot \mathbf{1}_{\left[a_\kappa + d_{\kappa,j} > a_n\right]} \cdot \mathbf{1}_{\left[j_\kappa^* = j \text{ and } i^*_{\kappa, j^*_\kappa} = i\right]}.$$

  end
  For each resource type $j \in [R]$, let $i^*_{n,j} = \arg\min_{i\in[C_j]} y_{n,j}^{(i)}$ and $y_{n,j} = \min_{i\in[C_j]} y_{n,j}^{(i)}$.
  For each resource type $j \in R$, determine

$$x_{n,j} = \arg\max_{x\in[0,1]} \Big\{x \cdot d_{n,j} - \int_{\eta=y_{n,j}}^{y_{n,j}+x} \phi_{1,j}(\eta) d\eta\Big\}.$$

  Determine $j_n^* = \arg\max_{j\in R} x_{n,j}\, \rho_{n,j}\, d_{n,j}$ and $x_n = x_{n,j_n^*}$.
  if $a_{j_n^*}^{(i^*_{n,j_n^*})} = 1$ **then**
    Allocate unit i^*_{n,j_n^*} of resource j_n^* to buyer n with probability $\dfrac{x_n}{1 - y_{n,j_n^*}}$.
    if *a successful allocation occurs* **then**
      $a_{j_n^*}^{(i^*_{n,j_n^*})} \leftarrow 0$.
    end
  end
end

More specifically, at the beginning, Algorithm 6 computes a set of static prices $p_j, \forall j \in [J]$, which are set randomly using the set of random seeds $\{s_j\}_{j\in[J]}$ sampled from the uniform distribution $\mathcal{U}(0,1)$. These prices remains unchanged for the entire horizon of `Matching-RR` problem where buyers arrive. At the arrival of each buyer n, the algorithm computes a set of values $y_{n,j}^{(i)}$, which is the probability that unit i of resource type j is allocated to a buyer arriving before buyer n and is still under rental at the time of buyer n's arrival. We refer to this value as the probabilistic utilization level of unit i of resource type j. Then, for each resource type j, the algorithm selects the unit $i_{n,j}^*$ with the lowest utilization level and sets the variable $y_{n,j}$ to be the utilization level of unit $i_{n,j}^*$.

Next, the algorithm sets R to be the subset of resource types for which buyer n's per-unit-time valuation is at least equal to the static price that was set for that resource type at the beginning of the horizon. Then, for each resource type j, the algorithm computes a variable $x_{n,j}$ in a manner similar to Algorithm 4 to handle the uncertainty regarding buyers' rental durations. At the final stage, to determine which resource type to allocate (i.e., matching) to buyer n, the algorithm selects the resource type j_n^* that will yield the highest expected value if a unit of that resource type is allocated to buyer n with probability x_{n,j_n^*}. Finally, the algorithm allocates unit $i_{n,j_n^*}^*$ of resource j_n^* to buyer n with probability x_n using the same limited correlation procedure as rDynamic-variable algorithm given in Algorithm 4.

6.3 Theoretical Guarantee of rHybrid-matching-RR

Using the randomized version of the LP-free certificate method in [GIU0], we obtain a set of constraints for the family of pricing functions $\{\phi_{1,j},\, \phi_{2,j}\}_{j\in[J]}$ to satisfy, ensuring that rHybrid-matching-RR achieves a specified level of competitiveness for the `Matching-RR` problem.

Proposition 6.1. *rHYBRID-MATCHING-RR (Algorithm 6) is*

$$\max_{j\in[J]}\left\{\left(1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)\cdot\alpha_j\right\}\text{-competitive}$$

for the `Matching-RR` *problem, provided that for each resource j:*

- *The pricing function $\phi_{1,j}$ is increasing and satisfies the following inequalities for all $d \in [d_{\min,j},\, d_{\max,j}]$:*

$$\int_{y_1}^{2y_1}\frac{2\alpha_j}{3}\,\phi_{1,j}(\eta)\,d\eta+\frac{\alpha_j}{3}\,d\left(\phi_{1,j}^{-1}(d)-2y_1\right)\geq d,\quad\forall\, y_1\in\left[0,\,\frac{\phi_{1,j}^{-1}(d)}{2}\right],\tag{6.2}$$

$$\frac{\alpha_j}{3}\,d\,\phi_{1,j}^{-1}(d)-\frac{\alpha_j}{3}\,y_2\left(d-\phi_{1,j}(y_2)\right)\geq d,\quad\forall\, y_2\in\left[0,\,\phi_{1,j}^{-1}(d)\right].\tag{6.3}$$

- *The pricing function $\phi_{2,j}$ is defined as*

$$\phi_{2,j}(y)=\rho_{\min,j}\cdot\exp\left(\left[1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right]y-1\right),\quad\forall\, y\in[0,1].$$

The proof of the above proposition can be found in Appendix E.1. For each resource type j, the constraints that the pricing function $\phi_{1,j}$ must satisfy—parameterized by α_j—are similar to those in Proposition 4.1. Each function $\phi_{1,j}$ is parameterized by a different α_j because the uncertainty range with respect to buyer's rental duration for each resource type, $[d_{\min,j}, d_{\max,j}]$, differs. Furthermore, the pricing functions $\{\phi_{2,j}\}_{j\in[J]}$ are designed to handle the uncertainty in the per-unit-time valuation of each resource j. The final competitive ratio of rHYBRID-MATCHING-RR is given by the maximum product over resource types of the terms α_j and $\left(1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)$, where α_j reflects the efficiency of $\phi_{1,j}$ in managing uncertainty regarding buyers' rental durations, and $\left(1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)$ reflects the efficiency of $\phi_{2,j}$ in managing uncertainty regarding buyers' per-unit-time valuations.

Corollary 6.1. *rHYBRID-MATCHING-RR (Algorithm 6) is*

$$3\cdot\max_{j\in[J]}\left\{\left(1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)\cdot\left(1+\ln\left(\frac{d_{\max,j}}{d_{\min,j}}\right)\right)\right\}\text{-competitive}$$

for the problem `Matching-RR` *if the set of pricing functions $\{\phi_{1,j},\phi_{2,j}\}_{j\in[J]}$ are given as follows:*

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

$$\phi_{2,j}(y)=\rho_{\min,j}\cdot\exp\left(\left[1+\ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right]y-1\right),\quad\forall y\in[0,1].$$

The above corollary follows by designing the set of pricing functions $\{\phi_{1,j}\}_{j\in[J]}$ that satisfy the constraints in Eqs. (6.2) and (6.3), in a manner similar to Corollary 4.1. It is worth noting that the proposed designs for each $\phi_{1,j}$ satisfy these constraints with $\alpha_j = 3 \cdot \left(1 + \ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)$; however, this value of α_j is not necessarily the lowest possible. To achieve pricing functions with tighter performance guarantees, one can adopt a numerical approach and solve the optimization problem in Eq. (4.8a), parameterized by $d_{\min,j}$ and $d_{\max,j}$.

6.4 Lower Bound for `Matching-RR` and Order-Optimality of rHybrid-matching-RR

We can prove the following lower-bound on the competitive-ratio of any online algorithm for `Matching-RR` problem using the representative function approach similar to that of Chapter 2.

Proposition 6.2. *No online algorithm, deterministic or randomized, can obtain a competitive ratio better than* $\max_{j\in[J]}\left(1 + \ln\left(\frac{d_{\max,j}}{d_{\min,j}}\right) + \ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right)\right)$ *for the* ***`Matching-RR`*** *problem.*

We prove the above proposition by designing a set of hard instances and analyzing the performance of the optimal online algorithm on these instances using the representative function approach. Further details of the proof can be found in Appendix E.2.

Building on the above lower bound result and Corollary 6.1, we argue that the competitive ratio of the rHybrid-matching-RR algorithm is order-optimal with respect to the fluctuation ratios in rental durations and per-unit-time valuations, i.e., $\frac{d_{\max,j}}{d_{\min,j}}$ and $\frac{\rho_{\max,j}}{\rho_{\min,j}}$ for each resource type independently.

6.5 Chapter Summary

In this chapter, we introduce the online edge-weighted matching problem with bounded weights and reusable resources (`Matching-RR`), a generalization of the problem studied in [SYH+23]. Unlike their setting, our model allows buyers to arrive at any point in time over a continuous horizon. This problem also generalizes the k-rental problems examined in Chapters 3 and 4.

We propose a randomized hybrid pricing algorithm, rHybrid-matching-RR, which integrates both static and dynamic pricing techniques to address uncertainties in buyer valuations and rental durations. Specifically, the algorithm first applies static randomized pricing to manage uncertainty in per-unit-time valuations, followed by dynamic pricing to adapt to variability in rental intervals. This hybrid approach enables the algorithm to effectively balance inventory across multiple resource types. Our theoretical analysis establishes a new lower bound on the competitive ratio and shows that rHybrid-matching-RR achieves order-optimal competitiveness, successfully addressing the challenges of online allocation with reusable resources.

It remains an open problem to design algorithms with tighter performance guarantees and to establish improved lower bounds for the performance of any online algorithm for the `Matching-RR` problem.

Chapter 7

Conclusions, Open Questions, and Future Work

In this thesis, we studied online resource allocation in settings that range from online selection under increasing marginal production costs (with single type of resource) to online matching with reusable and multi-type resources. More specifically, Chapter 2 investigated diseconomies of scale in online selection by proposing a randomized dynamic pricing scheme and establishing tight lower bounds. Chapters 3 and 4 focused on the online k-rental problem, first under a fixed rental duration and then under variable durations. Finally, Chapters 5 and 6 moved to multi-resource environments, showing that carefully designed static and dynamic pricing methods can achieve optimal in the first case and near-optimal competitive ratios even in the multi-resource setting where there is another layer of uncertainty compared to online selection problems regarding the total demand for different resource types.

There are some limitations in the results presented in this thesis that require further exploration. In particular, it remains an open question whether Algorithm 1, discussed in Chapter 2, is optimal for all $k \geq 3$. Although we conjecture that our randomized pricing mechanism is optimal for all $k \geq 1$, establishing or refuting its optimality for $k \geq 3$ requires a more refined analysis. Additionally, designing algorithms with tight performance guarantees for the problems studied in Chapter 4 and Chapter 6 remains a challenge. Future directions include developing improved pricing functions for resource allocation and devising stronger correlation schemes that can more effectively round the fractional solution than our current methods allow. Finally, since the results in Chapter 3, Chapter 4, and Chapter 6 are not incentive-compatible with respect to buyer valuations, an important direction for future work is to design online mechanisms that guarantee incentive compatibility in these scenarios.

Furthermore, our work raises broader questions in the field of online resource allocation. One intriguing direction is to explore posted-price mechanisms that incorporate correlation across different prices and to analyze their performance. We believe that the correlation scheme (k, d)-OCA, introduced in Chapter 3, could pave the way for developing such posted-price mechanisms. In addition, it would be valuable to study other metrics, such as risk and fairness, to ensure that these randomized pricing algorithms not only maximize efficiency but also yield more reliable and equitable outcomes. In particular, investigating the risk sensitivity of randomized pricing algorithms would extend our current risk-neutral analysis, offering deeper insights into performance under uncertainty.

In Chapters 3, 4, and 6, we employed a relax-and-round framework to design online algorithms for problems involving reusable resources. In the relaxation step, our pricing-based approach for generating fractional solutions proved effective, yielding strong performance guarantees. This outcome highlights the broader applicability of pricing-based methods to other related settings. In the rounding step, although our correlation-based techniques demonstrated strong performance in the studied problems, there remains considerable room for improvement. In particular, the development of more general and powerful correlation schemes could further enhance performance, expanding the applicability of the relax-and-round framework to increasingly complex online market scenarios.

Bibliography

[AAP93] Baruch Awerbuch, Yossi Azar, and Serge Plotkin. Throughput-competitive on-line routing. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 32–40. IEEE, 1993.

[AAZ16] Matthew. Andrews, Spyridon. Antonakopoulos, and Lisa. Zhang. Minimum-cost network design with (dis)economies of scale. *SIAM Journal on Computing*, 45(1):49–66, 2016.

[ABC$^+$16] Yossi Azar, Niv Buchbinder, TH Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, et al. Online algorithms for covering and packing problems with convex objectives. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 148–157. IEEE, 2016.

[AGKM11] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, page 1253–1264, USA, 2011. Society for Industrial and Applied Mathematics.

[Ala14] Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014.

[AS23] Ali Aouad and Daniela Saban. Online assortment optimization for two-sided matching platforms. *Management Science*, 69(4):2069–2087, 2023.

[BEY98] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[BGMS11] Avrim Blum, Anupam Gupta, Yishay Mansour, and Ankit Sharma. Welfare and profit maximization with production costs. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 77–86. IEEE, 2011.

[BMY15] Avrim Blum, Yishay Mansour, and Liu Yang. Online allocation and pricing with economies of scale. In *Web and Internet Economics: 11th International Conference, WINE 2015, Amsterdam, The Netherlands, December 9-12, 2015, Proceedings 11*, pages 159–172. Springer, 2015.

[BN09] Niv Buchbinder and Joseph (Seffi) Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

[BQ09] Michael O Ball and Maurice Queyranne. Toward robust revenue management: Competitive analysis of online booking. *Operations Research*, 57(4):950–963, 2009.

[BUCM12] Siddharth Barman, Seeun Umboh, Shuchi Chawla, and David Malec. Secretary problems with convex costs. In *International Colloquium on Automata, Languages, and Programming*, pages 75–87. Springer, 2012.

[CFH+19] Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Recent developments in prophet inequalities. *ACM SIGecom Exchanges*, 17(1):61–70, 2019.

[CST22] Ying Cao, Bo Sun, and Danny HK Tsang. Online network utility maximization: Algorithm, competitive analysis, and applications. *IEEE Transactions on Control of Network Systems*, 10(1):274–284, 2022.

[DFNS22] 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.

[DJ12] Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the 44th Annual ACM symposium on Theory of computing*, STOC '12, pages 137–144, 2012.

[DJK13] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. SODA '13, page 101–107, USA, 2013. Society for Industrial and Applied Mathematics.

[EFFS21] 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.

[EYFKT01] Ran El-Yaniv, Amos Fiat, Richard M Karp, and Gordon Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30:101–139, 2001.

[FGK96] Ulrich Faigle, Renate Garbe, and Walter Kern. Randomized online algorithms for maximizing busy time interval scheduling. *Computing*, 56(2):95–104, 1996.

[FHTZ22] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. *Journal of the ACM*, 69(6):1–35, 2022.

[FKMP09] Jon Feldman, Nitish Korula, Vahab Mirrokni, and Martin Pál. Online ad assignment with free disposal. In *International Conference on Internet and Network Economics (WINE '09)*, pages 374–385. Springer, 2009.

[FNS19] Yiding Feng, Rad Niazadeh, and Amin Saberi. Linear programming based online policies for real-time assortment of reusable resources. *SSRN Electronic Journal*, 01 2019.

[FSZ21] 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.

[Gar70] Martin Gardner. Mathematical games. *Scientific american*, 222(6):132–140, 1970.

[GG20] Shashank Goyal and Diwakar Gupta. The online reservation problem. *Algorithms*, 13(10), 2020.

[GGI+22] Xiao-Yue Gong, Vineet Goyal, Garud N. Iyengar, David Simchi-Levi, Rajan Udwani, and Shuangyu Wang. Online assortment optimization with reusable resources. *Management Science*, 68(7):4772–4785, 2022.

[GIU0] Vineet Goyal, Garud Iyengar, and Rajan Udwani. Asymptotically optimal competitive ratio for online allocation of reusable resources. *Operations Research*, 0(0):null, 0.

[GI02] 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.

[GL16] Diwakar Gupta and Fei Li. Reserve driver scheduling. *IIE Transactions*, 48(3):193–204, 2016.

[GMM18] Anupam Gupta, Ruta Mehta, and Marco Molinaro. Maximizing profit with convex costs in the random-order model. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, 2018.

[GNR14] Negin Golrezaei, Hamid Nazerzadeh, and Paat Rusmevichientong. Real-time optimization of personalized assortments. *Management Science*, 60(6):1532–1551, 2014.

[GS21] Anupam Gupta and Sahil Singla. *Random-Order Models. Chapter 11 – Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.

[HC23] Tianming Huo and Wang Chi Cheung. Online reusable resource assortment planning with customer-dependent usage durations. SSRN Working Paper, July 9 2023.

[HK19] Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: A primal dual approach. *Games and Economic Behavior*, 118:648–667, 2019.

[HS24] Erfan Hajihashemi and Yanning Shen. Multi-model ensemble conformal prediction in dynamic environments. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[HZZ24] Zhiyi Huang, Qiankun Zhang, and Yuhao Zhang. Adwords in a panorama. *SIAM Journal on Computing*, 53(3):701–763, 2024.

[JLL+23] Jianhao Jia, Hao Li, Kai Liu, Ziqi Liu, Jun Zhou, Nikolai Gravin, and Zhihao Gavin Tang. Online resource allocation in markov chains. In *Proceedings of the ACM Web Conference 2023*, pages 3498–3507, 2023.

[JLTZ21a] 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.

[JLTZ21b] Zhihao Jiang, Pinyan Lu, Zhihao Gavin Tang, and Yuhao Zhang. Online selection problems against constrained adversary. In *International Conference on Machine Learning*, pages 5002–5012. PMLR, 2021.

[JSBT25] 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.

[KFV15] A. Gürhan Kök, Marshall L. Fisher, and Ramnath Vaidyanathan. *Assortment Planning: Review of Literature and Industry Practice*, pages 175–236. Springer US, Boston, MA, 2015.

[KVV90] 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.

[LPS09] Julian Lorenz, Konstantinos Panagiotou, and Angelika Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

[LT94] Richard J. Lipton and Andrew Tomkins. Online interval scheduling. In *ACM-SIAM Symposium on Discrete Algorithms*, 1994.

[Luc17] Brendan Lucier. An economic view of prophet inequalities. *ACM SIGecom Exchanges*, 16(1):24–47, 2017.

[M+13] Aranyak Mehta et al. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

[Ma24] Will Ma. Randomized rounding approaches to online allocation, sequencing, and matching, 2024.

[MSL20] 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.

[OP12] Ozalp Ozer and Robert Phillips. *The Oxford Handbook of Pricing Management*. Oxford University Press, 2012.

[SC84] Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, pages 1213–1216, 1984.

[Sek17] Shreyas Sekar. Posted pricing sans discrimination. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 388–394, 2017.

[SJTB24] 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.

[SYH+22] 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*, 6(3):1–32, 2022.

[SYH+23] Bo Sun, Lin Yang, Mohammad Hajiesmaili, Adam Wierman, John C.S. Lui, Don Towsley, and Danny H.K. Tsang. The online knapsack problem with departures. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '23, page 59–60, New York, NY, USA, 2023. Association for Computing Machinery.

[TSLG+20] 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*, 4(2):1–46, 2020.

[TYBLG23] 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.

[ZCL08] 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.

[ZHW$^+$15] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau. Online auctions in iaas clouds: Welfare and profit maximization with server costs. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '15)*, pages 3–15, 2015.

[ZLW17] Zijun Zhang, Zongpeng Li, and Chuan Wu. Optimal posted prices for online cloud resource allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(1):1–26, 2017.

Appendix A

Proofs of Chapter 2

A.1 Proof of Lemma 2.1

The monotonicity of each function ψ_i follows from the fact that if the random variable $\Psi_i(v)$ is realized to be equal to one for some v, then for all $v' > v$, $\Psi_i(v')$ must also be equal to one (based on the definition of Ψ_i). Since $\psi_i(v)$ and $\psi_i(v')$ represent the expected values of $\Psi_i(v)$ and $\Psi_i(v')$, respectively, this property ensures that each ψ_i is increasing. Furthermore, by the definition of the state variables q_i and q_{i+1}, whenever $\Psi_{i+1}(v) = 1$, the allocation must have reached at least $i+1$ units, which implies $\Psi_i(v) = 1$. Consequently, it follows that $\psi_i(v) \geq \psi_{i+1}(v)$.

A.2 Proof of Proposition 2.1

For any randomized algorithm ALG, let $D(v_{\min})$ denote the number of units that ALG allocates under the instance $\mathcal{I}^{(\epsilon)}_{v_{\min}}$ (i.e., the instance $\mathcal{I}^{(\epsilon)}$ by the end of stage-$v_{\min}$). Thus, $D(v_{\min})$ is a random variable taking values from 0 to k. Based on definition of $D(v_{\min})$, $\texttt{ALG}(\mathcal{I}^{(\epsilon)}_{v_{\min}})$ can be computed as follows:

$$\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) = \mathbb{E}\left[D(v_{\min}) \cdot v_{\min} - \sum_{i=1}^{D(v_{\min})} c_i\right],$$

where the expectation is taken with respect to the randomness of $D(v_{\min})$ (the distribution depends on the randomness of the algorithm ALG). Let the indicator function $\mathbf{1}_{\{D(v_{\min})=j\}} = 1$ if ALG allocates exactly j units at the end of stage-$v_{\min}$, and $\mathbf{1}_{\{D(v_{\min})=j\}} = 0$ otherwise. Based on definition of the random variables $\{\Psi_i(v_{\min})\}_{\forall i\in[k]}$, we argue that:

$$\mathbf{1}_{\{D(v_{\min})=j\}} = \Psi_j(v_{\min}) - \Psi_{j+1}(v_{\min}), \qquad 1 \le j \le k. \tag{A.1}$$

Here, $\Psi_{k+1}(v_{\min}) = 0$ always holds. To see why Eq. (A.1) is true, consider the case where the random variable $D(v_{\min}) = j$, then:

$$\Psi_i(v_{\min}) = 1, \quad \forall i \le j,$$
$$\Psi_i(v_{\min}) = 0, \quad \forall i > j.$$

From the equation above, we can observe that when the indicator function $\mathbf{1}_{\{D(v_{\min})=j\}} = 1$, $\Psi_{j+1}(v_{\min}) - \Psi_j(v_{\min}) = 1$ holds. For the case when $\mathbf{1}_{\{D(v_{\min})=j\}} = 0$, if $D(v_{\min}) < j$, then $\Psi_j(v_{\min}) = \Psi_{j+1}(v_{\min}) = 0$ and $\mathbf{1}_{\{D(v_{\min})=j\}} = \Psi_j(v_{\min}) - \Psi_{j+1}(v_{\min})$ follows. For the case $D(v_{\min}) > j$, the two equations $\Psi_j(v_{\min}) = \Psi_{j+1}(v_{\min}) = 1$ and $\mathbf{1}_{\{D(v_{\min})=j\}} = \Psi_j(v_{\min}) - \Psi_{j+1}(v_{\min})$ again follow. As a result, $\texttt{ALG}(\mathcal{I}^{(\epsilon)}_{v_{\min}})$ can be computed as follows:

$$\begin{aligned}
\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) &= \mathbb{E}\left[D(v_{\min}) \cdot v_{\min} - \sum_{i=1}^{D(v_{\min})} c_i\right] \\
&= \sum_{j=1}^{k} \mathbb{E}\left[\mathbf{1}_{\{D(v_{\min})=j\}}\right] \cdot \left(j \cdot v_{\min} - \sum_{i=1}^{j} c_i\right) \\
&= \sum_{j=1}^{k} \mathbb{E}\left[\Psi_j(v_{\min}) - \Psi_{j+1}(v_{\min})\right] \cdot \left(j \cdot v_{\min} - \sum_{i=1}^{j} c_i\right) \\
&= \sum_{j=1}^{k} \left(\psi_j(v_{\min}) - \psi_{j+1}(v_{\min})\right) \cdot \left(j \cdot v_{\min} - \sum_{i=1}^{j} c_i\right) \\
&= \sum_{j=1}^{k} \psi_j(v_{\min}) \cdot (v_{\min} - c_j) - \psi_{k+1}(v_{\min}) \cdot \left(k \cdot v_{\min} - \sum_{i=1}^{k} c_i\right) \\
&= \sum_{j=1}^{k} \psi_j(v_{\min}) \cdot (v_{\min} - c_j).
\end{aligned}$$

Now, let us compute the objective of the α-competitive algorithm at the end of stage-v, $\forall v \in V^{(\epsilon)}$, such that $v = v_{\min} + m \cdot \epsilon$. Let the random variable $X_i(v)$ be the value obtained from allocating the i-th unit of the item at the end of some stage-$v \in V^{(\epsilon)}$. It follows that

$$\begin{aligned}
&\mathbb{E}[X_i(v) - c_i] \\
&= \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \mathbb{E}\left[\sum_{j=1}^{m}(v_{\min} + j \cdot \epsilon - c_i) \cdot \Big(\Psi_i(v_{\min} + j \cdot \epsilon) - \Psi_i(v_{\min} + (j-1) \cdot \epsilon)\Big)\right] \\
&= \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \sum_{j=1}^{m}(v_{\min} + j \cdot \epsilon - c_i) \cdot \mathbb{E}\left[\Psi_i(v_{\min} + j \cdot \epsilon) - \Psi_i(v_{\min} + (j-1) \cdot \epsilon)\right] \\
&= \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \sum_{j=1}^{m}(v_{\min} + j \cdot \epsilon - c_i) \cdot \left(\psi_i(v_{\min} + j \cdot \epsilon) - \psi_i(v_{\min} + (j-1) \cdot \epsilon)\right).
\end{aligned}$$

where the first equality follows because if the i-th unit is allocated at some stage $v_{\min} + j \cdot \epsilon$, then the algorithm must have sold at least i units of the item by the end of $v_{\min} + j \cdot \epsilon$, leading to $\Psi_i(v_{\min} + j \cdot \epsilon) = 1$. Additionally, if the i-th unit is allocated at stage $v_{\min} + j \cdot \epsilon$, then at stage $v_{\min} + (j-1) \cdot \epsilon$, the algorithm must have allocated fewer than i units, indicating that $\Psi_i(v_{\min} + (j-1) \cdot \epsilon) = 0$. Putting together the above results, it follows that:

$$\begin{aligned}
&\mathtt{ALG}\left(\mathcal{I}_v^{(\epsilon)}\right) \\
&= \sum_{i=1}^{k} \mathbb{E}[X_i(v) - c_i] \\
&= \sum_{i=1}^{k} \Big[\psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \\
&\quad \sum_{j=1}^{m}(v_{\min} + j \cdot \epsilon - c_i) \cdot \Big(\psi_i(v_{\min} + j \cdot \epsilon) - \psi_i(v_{\min} + (j-1) \cdot \epsilon)\Big)\Big], \\
&= \mathtt{ALG}\left(\mathcal{I}_{v_{\min}}^{(\epsilon)}\right) + \sum_{i=1}^{k}\sum_{j=1}^{m}(v_{\min} + j \cdot \epsilon - c_i) \cdot \Big(\psi_i(v_{\min} + j \cdot \epsilon) \\
&\qquad - \psi_i(v_{\min} + (j-1) \cdot \epsilon)\Big), \qquad \forall m \in \left\{1, \dots, \lfloor \frac{v_{\max} - v_{\min}}{\epsilon} \rfloor\right\}.
\end{aligned}$$

Proposition 2.1 thus follows.

A.3 Proof of Lemma 2.2

Based on Proposition 2.1, for any online algorithm ALG, we have:

$$\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) = \sum_{i=1}^{k} \psi_i^{(v_{\min})} \cdot (v_{\min} - c_i),$$

$$\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}+j\cdot\epsilon}\right) = \texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) + \sum_{i=1}^{k}\sum_{m=1}^{j}\Bigg((v_{\min}+m\cdot\epsilon)\cdot\Big(\psi_i(v_{\min}+m\cdot\epsilon) - \psi_i(v_{\min}+m\cdot\epsilon-\epsilon)\Big)\Bigg), \forall j = 1,2,\ldots,\left\lfloor\frac{v_{\max}-v_{\min}}{\epsilon}\right\rfloor.$$

As $\epsilon \to 0$, following the Riemann summation, it follows that:

$$\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v}\right) = \texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) + \sum_{i=1}^{k}\int_{\eta=v_{\min}}^{v}(\eta - c_i)\cdot\Big[\psi_i(\eta) - \psi_i(\eta - d\eta)\Big], \forall v \in [v_{\min}, v_{\max}].$$

Based on above, the set of functions $\{\psi_i\}_{i\in[k]}$ should be defined over the range $[v_{\min}, v_{\max}]$.

In the next step, we prove the existence of a set of functions $\{\psi_i\}_{i\in[k]}$ that are continuous on the interval $[v_{\min}, v_{\max}]$. For now, assume this claim holds. Then, we have

$$\begin{aligned}\texttt{ALG}\big(\mathcal{I}^{(\epsilon)}_{v}\big) &= \texttt{ALG}\big(\mathcal{I}^{(\epsilon)}_{v_{\min}}\big) + \sum_{i=1}^{k}\int_{v_{\min}}^{v}(\eta - c_i)\cdot\Big[\psi_i(\eta)-\psi_i(\eta-d\eta)\Big]\,d\eta\\ &= \texttt{ALG}\big(\mathcal{I}^{(\epsilon)}_{v_{\min}}\big) + \sum_{i=1}^{k}\int_{v_{\min}}^{v}(\eta - c_i)\,d\psi_i(\eta).\end{aligned}$$

Following from Eq. (2.4), if there exists an α-competitive algorithm, then there should exists a set of functions $\{\psi_i\}_{i\in[k]}$ such that:

$$\begin{aligned}\texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v}\right) &= \texttt{ALG}\left(\mathcal{I}^{(\epsilon)}_{v_{\min}}\right) + \sum_{i=1}^{k}\int_{\eta=v_{\min}}^{v}(\eta - c_i)\cdot d\psi(\eta)\\ &\geq \frac{1}{\alpha}\cdot\left(kv - \sum\nolimits_{i=1}^{k} c_i\right), \qquad \forall v \in [v_{\min}, v_{\max}].\end{aligned}$$

Now, let us get back to prove that a set of functions $\{\psi_i\}_{i\in[k]}$ exists corresponding to some online algorithm, that all these functions are continuous within the range $[v_{\min}, v_{\max}]$. Let ALG be an α-competitive algorithm. For some $v \in (v_{\min}, v_{\max})$ and $i \in [k]$, let the function $\psi_i(.)$ corresponding to ALG be non-continuous at v. Let $\lim_{x\to v^-}\psi_i(v) = \nu$ and $\psi_i(v) = \lim_{x\to v^+}\psi_i(v) = \nu+\delta$, for some $\delta > 0$. Then the algorithm must be selling at least in expectation a δ-fraction of the i-th unit to the buyers with valuation v in instance $\mathcal{I}$. Conversely, for $\texttt{ALG}$ to be α-competitive, the expected objective of the algorithm before the arrival of buyers with valuation v, $\texttt{ALG}(\mathcal{I}^{(\epsilon)}_{v^-})$, must be at least equal to $\frac{1}{\alpha} \cdot \texttt{OPT}(\mathcal{I}^{(\epsilon)}_{v^-}) = \frac{1}{\alpha} \cdot \texttt{OPT}(\mathcal{I}^{(\epsilon)}_{v})$, where $\texttt{OPT}(\mathcal{I}^{(\epsilon)}_{v})$ denotes the objective value of the offline optimal algorithm on the hard instance $\mathcal{I}^{(\epsilon)}$ up to the end of stage-v. It can be seen that selling in expectation at least a δ fraction of the i-th unit is unnecessary and $\texttt{ALG}$ could save this fraction of the unit and sell it to buyers with higher valuations. In other words, we can construct another online algorithm, say $\widehat{\texttt{ALG}}$, that follows $\texttt{ALG}$ up to the arrival of buyers with valuation v, but sells the δ-fraction of the i-th unit to buyers with valuation strictly greater than v instead. It is easy to see that $\widehat{\texttt{ALG}}$ will obtain a better objective value with its $\hat{\psi}_i$ being continuous at v. Lemma 2.1 follows by repeating the same process for any other discontinuous point of $\psi_i(v)$.

A.4 Proof of Lemma 2.3

For any $v \in [v_{\min}, v_{\max}]$, let us define C_v as follows:

$$C_v = C_{v_{\min}} + \sum_{i=1}^{k} \int_{\eta=v_{\min}}^{v} (\eta - c_i) d\psi_i^{\alpha}(\eta), \quad \forall v \in (v_{\min}, v_{\max}],$$
$$C_{v_{\min}} = \sum_{i=1}^{k} \psi_i^{\alpha}(v_{\min}) \cdot (v_{\min} - c_i).$$

To prove Lemma 2.3, we need to first prove the feasibility of $\{\psi_i^{(\alpha)}\}_{\forall i \in [k]}$, namely, C_v is greater than $\frac{1}{\alpha} \cdot (k \cdot v - \sum_i c_i)$ for all $v \in [v_{\min}, v_{\max}]$.

For some $v \in [v_{\min}, v_{\max}]$, based on the definition of $\chi^{\alpha}(v)$ in Eq. (2.6), there exist a set of functions $\{\psi_i(v)\}_{\forall i \in [k]}$ that satisfy Eq. (2.5) and in the meanwhile, for some arbitrary small value ϵ, we have:

$$\chi^{\alpha}(v) + \epsilon \geq \sum_{i=1}^{k} \psi_i(v). \tag{A.2}$$

Next, using integration by parts, we have

$$
\begin{aligned}
C_v &= C_{v_{\min}} + \sum_{i=1}^{k} \int_{\eta=v_{\min}}^{v} (\eta - c_i) d\psi_i^{\alpha}(\eta) \\
&= C_{v_{\min}} + \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot (v - c_i) - \sum_{i=1}^{k} \psi_i^{\alpha}(v_{\min}) \cdot (v_{\min} - c_i) - \int_{\eta=v_{\min}}^{v} \left(\sum_{i=1}^{k} \psi_i^{\alpha}(\eta) \right) d\eta \\
&= \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot (v - c_i) - \int_{\eta=v_{\min}}^{v} \left(\sum_{i=1}^{k} \psi_i^{\alpha}(\eta) \right) d\eta \\
&= v \cdot \left(\sum_{i=1}^{k} \psi_i^{\alpha}(v) \right) - \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i - \int_{\eta=v_{\min}}^{v} \left(\sum_{i=1}^{k} \psi_i^{\alpha}(\eta) \right) d\eta \\
&= v \cdot \chi^{\alpha}(v) - \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i - \int_{\eta=v_{\min}}^{v} \chi^{\alpha}(v) d\eta,
\end{aligned}
$$

where the last equality follows the definition of $\{\psi_i^{\alpha}\}_{\forall i \in [k]}$ in Eq. (2.7). Thus, we have

$$
\begin{aligned}
C_v &= v \cdot \chi^{\alpha}(v) - \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i - \int_{\eta=v_{\min}}^{v} \chi^{\alpha}(v) \cdot d\eta, \\
&\geq v \cdot \sum_{i=1}^{k} \psi_i(v) - v \cdot \epsilon - \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i - \int_{\eta=v_{\min}}^{v} \chi^{\alpha}(v) \cdot d\eta, \\
&\geq v \cdot \sum_{i=1}^{k} \psi_i(v) - \int_{\eta=v_{\min}}^{v} \left(\sum_{i=1}^{k} \psi_i(\eta) \right) \cdot d\eta - \sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i - v \cdot \epsilon, \qquad \text{(A.3)}
\end{aligned}
$$

where the first inequality follows Eq. (A.2) and the second inequality directly follows the definition of $\chi^{\alpha}(v)$ (recall that $\chi^{\alpha}(v) \leq \sum_{i=1}^{k} \psi_i(v)$ holds for all $v \in [v_{\min}, v_{\max}]$).

By the definition of $\{\psi_i^{\alpha}\}_{\forall i \in [k]}$, we have $\sum_{i \in [k]} \psi_i^{\alpha}(v) = \chi^{\alpha}(v)$. Putting together the inequality $\chi^{\alpha}(v) \leq \sum_{i=1}^{k} \psi_i(v)$ and the fact that productions costs are increasing, we have

$$
\sum_{i=1}^{k} \psi_i^{\alpha}(v) \cdot c_i \leq \sum_{i=1}^{k} \psi_i(v) \cdot c_i.
$$

Putting together the above inequality and the right-hand-side of Eq. (A.3), it follows that:

$$
\begin{aligned}
C_v &\geq v \cdot \sum_{i=0}^{k-1} \psi_i(v) - \sum_{i=0}^{k-1} \int_{\eta=v_{\min}}^{v} \psi_i(\eta) \cdot d\eta - \sum_{i=0}^{k-1} \psi_i(v) \cdot c_{i+1} - v \cdot \epsilon \\
&\geq \texttt{ALG}\left(\mathcal{I}_v^{(\epsilon)} \right) - v \cdot \epsilon,
\end{aligned}
$$

where ALG is the online algorithm corresponding to the set of allocation functions $\{\psi_i\}_{\forall i \in [k]}$ and recall that $\texttt{ALG}(\mathcal{I}_v^{(\epsilon)})$ is defined as follows:

$$\texttt{ALG}\left(\mathcal{I}_{v_{\min}}^{(\epsilon)}\right) = \sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_i),$$
$$\texttt{ALG}\left(\mathcal{I}_{v}^{(\epsilon)}\right) = \texttt{ALG}\left(\mathcal{I}_{v_{\min}}^{(\epsilon)}\right) + \sum_{i=1}^{k} \int_{\eta=v_{\min}}^{v} (\eta - c_i) d\psi_i(\eta), \quad \forall v \in [v_{\min}, v_{\max}].$$

Since $\{\psi_i\}_{\forall i\in[k]}$ satisfy Eq. (2.5), it follows that

$$C_v \geq \texttt{ALG}\left(\mathcal{I}_{v}^{(\epsilon)}\right) - v \cdot \epsilon$$
$$\geq \frac{1}{\alpha} \cdot \left(k \cdot v - \sum_{i=1}^{k} c_i\right) - v \cdot \epsilon, \quad \forall v \in [v_{\min}, v_{\max}].$$

By setting $\epsilon \to 0$, it follows that

$$C_v \geq \frac{1}{\alpha} \cdot \left(k \cdot v - \sum_{i=1}^{k} c_i\right), \quad \forall v \in [v_{\min}, v_{\max}].$$

To complete the proof of Lemma 2.3, we also need to prove that the above inequality holds as an equality for the set of functions $\{\psi_i^\alpha\}_{\forall i\in[k]}$. This can be proved by contradiction. Suppose that at some point $v \in [v_{\min}, v_{\max}]$, the above equality does not hold, then there must exist another set of feasible functions, say $\{\hat{\psi}_i\}_{\forall i\in[k]}$, induced by a new algorithm, say $\widehat{\texttt{ALG}}$, that satisfy Eq. (2.5) and

$$\sum_{i=1}^{k} \hat{\psi}_i(v) < \sum_{i=1}^{k} \psi_i^\alpha(v).$$

We argue that the new set of functions $\{\hat{\psi}_i\}_{\forall i\in[k]}$ will allocate a smaller fraction of its total units to buyers in $\mathcal{I}^{(\epsilon)}$ arriving at or before stage-v compared to $\{\psi_i^\alpha\}_{\forall i\in[k]}$. However, by still following the allocation functions $\{\psi_i^\alpha\}_{\forall i\in[k]}$, $\widehat{\texttt{ALG}}(\mathcal{I}_v^{(\epsilon)})$ will be exactly equal to $\frac{1}{\alpha}(k \cdot v - \sum_{i=1}^{k} c_i)$. Given the definition of $\{\psi_i^\alpha\}_{\forall i\in[k]}$, we have $\sum_{i=1}^{k} \psi_i^\alpha(v) = \chi^\alpha(v)$, meaning that $\sum_{i=1}^{k} \hat{\psi}_i(v) < \chi^\alpha(v)$. However, this contradicts the definition of $\chi^\alpha(v)$. We thus complete the proof of Lemma 2.3.

A.5 Proof of Proposition 2.2

From Lemma 2.3, we know that $\{\psi_i^\alpha(v)\}_{\forall i\in[k]}$ satisfy Eq. (2.5) with an equality. Therefore, the set of allocation functions $\{\psi_i^\alpha(v)\}_{\forall i\in[k]}$ is a solution to the following system of equations:

$$\begin{aligned}&\sum_{i=1}^{k}\psi_i^\alpha(v_{\min})\cdot(v_{\min}-c_i)+\sum_{i=1}^{k}\int_{\eta=v_{\min}}^{v}(\eta-c_i)d\psi_i^\alpha(\eta)\\ =&\ \frac{1}{\alpha}\cdot(k\cdot v-\sum_i c_i),\quad \forall i\in[k],v\in[v_{\min},v_{\max}].\end{aligned}\tag{A.4}$$

Also, based on Lemma 2.4, we argue that if the value of the function $\psi_i^\alpha(v)$ is changing at some value $v\in[v_{\min},v_{\max}]$ (i.e., $d\psi_i^\alpha(v)\neq 0$), then the value of all the functions $\{\psi_j^*(v)\}_{\forall j\in[i-1]}$ are equal to one, and all the functions in the set $\{\psi_j^*(v)\}_{j>i}$ are equal to zero. Based on this property, we can assign an interval $[\ell_i,u_i]$ to each $\psi_i^\alpha(v)$. In the interval of $[\ell_i,u_i]$, only the value of ψ_i^α changes while the other functions $\{\psi_j^\alpha\}_{\forall j\neq i}$ in that interval are fixed to be one or zero. Additionally, the following relation exists between the start and end points of these intervals:

$$v_{\min}=\ell_1\le u_1=\ell_2\le u_2\le\cdots\le\ell_k\le u_k=v_{\max}.$$

To satisfy the equality $\sum_{i\in[k]}\psi_i^\alpha(v_{\min})\cdot(v_{\min}-c_i)=\frac{1}{\alpha}\cdot(k\cdot v_{\min}-\sum_i c_i)$, the set of functions $\{\psi_i^\alpha(v)\}_{\forall i\in[\underline{k}-1]}$ should be equal to one at the point $v=v_{\min}$. Thus, the explicit design of the functions $\{\psi_i^\alpha\}_{\forall i\in[\underline{k}-1]}$ is as follows:

$$\psi_i^{(\alpha)}(v)=1,\quad i=1,\dots,\underline{k}-1.$$

In the case that $\sum_{i\in[\underline{k}]}v_{\min}-c_i<\frac{1}{\alpha}\cdot(k\cdot v_{\min}-\sum_i c_i)$, to satisfy $\sum_{i\in[k]}\psi_i^\alpha(v_{\min})\cdot(v_{\min}-c_i)=\frac{1}{\alpha}\cdot(k\cdot v_{\min}-\sum_i c_i)$, we need to have:

$$\psi_{\underline{k}}^\alpha(v_{\min})=\frac{\sum_{i\in[\underline{k}-1]}(v_{\min}-c_i)-\frac{1}{\alpha}\cdot\sum_{i\in[k]}(v_{\min}-c_i)}{v_{\min}-c_{\underline{k}}}=\xi.$$

Since for all $v\in[\ell_{\underline{k}},u_{\underline{k}}]$ with $\ell_{\underline{k}}=v_{\min}$, only the value of $\psi_{\underline{k}}^\alpha(v)$ changes (i.e., $d\psi_i^\alpha(v)=0$ for all $i\neq\underline{k}$), it follows that:

$$\sum_{i=1}^{k}\psi_i^\alpha(v_{\min})\cdot(v_{\min}-c_i)+\int_{\eta=v_{\min}}^{v}(\eta-c_{\underline{k}})d\psi_{\underline{k}}^*(\eta)=\frac{1}{\alpha}\cdot(k\cdot v-\sum_i c_i),\qquad\forall v\in[\ell_{\underline{k}},u_{\underline{k}}].$$

Taking derivative w.r.t. v from both sides of the equation above, we have

$$(v - c_{\underline{k}}) \cdot d\psi^*_{\underline{k}}(v) = \frac{k}{\alpha}.$$

Solving the above differential equation leads to

$$\psi^*_{\underline{k}}(v) = \frac{k}{\alpha} \cdot \ln(v - c_{\underline{k}}) + Q, \quad \forall v \in [\ell_{\underline{k}}, u_{\underline{k}}],$$

where Q is a constant. To find Q, since $\psi^*_{\underline{k}}(v_{\min}) = \xi$, it follows that $Q = \xi - \frac{k}{\alpha} \cdot \ln(v_{\min} - c_{\underline{k}})$. As a result, the explicit design of the function $\psi^{\alpha}_{\underline{k}}$ is as follows:

$$\psi^{(\alpha)}_{\underline{k}}(v) = \begin{cases} \xi + \frac{k}{\alpha} \cdot \ln\left(\frac{v - c_{\underline{k}}}{v_{\min} - c_{\underline{k}}}\right) & v \in [v_{\min}, u_{\underline{k}}], \\ 1 & v > u_{\underline{k}}. \end{cases}$$

To obtain the value of $u_{\underline{k}}$, we set $\psi^*_{\underline{k}}(u_{\underline{k}}) = 1$ (the function $\psi^*_{\underline{k}}$ reaches its maximum). Consequently, it follows that:

$$u_{\underline{k}} = (v_{\min} - c_{\underline{k}}) \cdot e^{\frac{\alpha}{k} \cdot (1 - \xi)} + c_{\underline{k}}.$$

Using the same procedure as what has been applied to $\psi^{\alpha}_{\underline{k}}$, for all the other functions $\{\psi^{\alpha}_i(v)\}_{\forall i > \underline{k}}$, we have

$$\begin{aligned} &\sum_{j=1}^{k} \psi^*_j(v_{\min}) \cdot (v_{\min} - c_j) + \sum_{j=1}^{k} \int_{\eta = v_{\min}}^{v} (\eta - c_j) d\psi^{\alpha}_i(v) \\ = &\sum_{j=1}^{k} \psi^*_j(v_{\min}) \cdot (v_{\min} - c_j) + \int_{\eta = v_{\min}}^{v} (\eta - c_i) d\psi^{\alpha}_i(v), \quad \forall v \in [\ell_i, u_i]. \end{aligned}$$

Taking derivative w.r.t. v from both sides of the equation above, it follows that:

$$(v - c_i) \cdot d\psi^{\alpha}_i(v) = \frac{k}{\alpha}.$$

Solving the above differential equation leads to

$$\psi^{\alpha}_i(v) = \frac{k}{\alpha} \cdot \ln(v - c_i) + \hat{Q}, \quad \forall v \in [\ell_i, u_i].$$

Since $\psi^*(\ell_i) = 0$, we have $\hat{Q} = -\ln(\ell_i - c_i)$. The explicit design of the function ψ_i^α is thus as follows:

$$\psi_i^{(\alpha)}(v) = \begin{cases} 0 & v \leq \ell_i, \\ \frac{k}{\alpha} \cdot \ln\left(\frac{v - c_i}{\ell_i - c_i}\right) & v \in [\ell_i, u_i], \quad i = \underline{k}+1, \ldots, k-1 \\ 1 & v \geq u_i. \end{cases}$$

For the function ψ_k^α, since it is the last function, it follows that:

$$\psi_k^{(\alpha)}(v) = \begin{cases} 0 & v \leq \ell_k, \\ \frac{k}{\alpha} \cdot \ln\left(\frac{v - c_k}{\ell_k - c_k}\right) & v \in [\ell_k, v_{\max}], \end{cases}$$

By setting $\psi^*(u_i) = 1$, it follows that:

$$u_i = (\ell_i - c_i) \cdot e^{\frac{\alpha}{k}} + c_i, \quad \underline{k}+1 \leq i \leq k.$$

Putting everything together, Proposition 2.2 follows.

A.6 Full Proof of Theorem 2.2

In this subsection, we provide a complete proof of Theorem 5. We begin by introducing several important notations and lemmas. Then, we break the problem into two independent subproblems based on the buyers' valuations in some arbitrary arrival instance $\mathcal{I}$. For each case, we proceed to show how to upper bound $\texttt{OPT}(\mathcal{I})$, the objective of the offline optimal algorithm on $\mathcal{I}$. We then proceed to lower bound the expected performance of r-Dynamic-cost on that instance, $\texttt{ALG}(\mathcal{I})$. Ultimately, we combine everything and obtain a performance guarantee for r-Dynamic-cost under all adversarially chosen instances of `kSelection-cost` for that subproblem.

Notations and Definitions

Consider an arbitrary arrival instance $\mathcal{I} = \{v_n\}_{n \in [N]}$. Recall that the random price vector $\mathbf{P} = \{P_1, \cdots, P_k\}$ is generated using the pricing functions $\{\phi_i\}_{\forall i \in [k]}$ at the beginning of r-Dynamic-cost. In the following, we will refer to Algorithm 1 as r-Dynamic-cost($\mathbf{P}$) to indicate that the algorithm is executed with the random price vector being realized as $\mathbf{P}$. Based on the design of $\{\phi_i\}_{\forall i \in [k]}$ in Theorem 2.2, the first $k^* - 1$ prices in $\mathbf{P}$ are all $v_{\min}$'s (i.e., $P_1 = \cdots = P_{k^*-1} = v_{\min}$), the k^*-th price P_{k^*} is a random variable within $[v_{\min}, U_{\underline{k}^*}]$, and for all $i \in \{k^*+1, \cdots, k\}$, we have $P_i \in [L_i, U_i]$ (recall that P_i is also a random variable). Here, the values of $\underline{k}^*$ and $\{[L_i, U_i]\}_{\forall i}$ are all defined in Theorem 2.2.

Let $\mathcal{P}$ denote the support of all possible values of the random price vector $\mathbf{P}$:

$$\mathcal{P} = \{v_{\min}\}^{\underline{k}^*-1} \times [v_{\min}, U_{\underline{k}^*}] \times \prod_{i \in \{\underline{k}^*+1, \cdots, k\}} [L_i, U_i].$$

Given a price vector realization $\mathbf{P} \in \mathcal{P}$, let $W(\mathbf{P})$ represent the total number of items allocated by r-Dynamic-cost($\mathbf{P}$) under the input instance $\mathcal{I}$. Since $\mathbf{P}$ is a random variable, $W(\mathbf{P})$ is also a random variable. For clarity, we will sometimes omit the price vector and refer to it simply as W whenever the context is clear.

Let ω denote the maximum value in the support of the random variable W (i.e., ω is the maximum possible value of $W(\mathbf{P})$ for all $\mathbf{P} \in \mathcal{P}$). Thus, ω is a deterministic value that depends only on the input instance $\mathcal{I}$. Furthermore, let $\boldsymbol{\pi} \in \mathcal{P}$ be a price vector such that r-Dynamic-cost($\boldsymbol{\pi}$) allocates the ω-th item earlier than any other price vector in the set $\mathcal{P}$. That is, for all $\mathbf{P} \in \mathcal{P}$, r-Dynamic-cost($\mathbf{P}$) allocates the ω-th item no earlier than that of r-Dynamic-cost($\boldsymbol{\pi}$).

Let us define the set $\{(\nu_i, \tau_i)\}_{\forall i \in [\omega]}$ so that τ_i is the arrival time of the buyer in the instance $\mathcal{I}$ to whom r-Dynamic-cost($\boldsymbol{\pi}$) allocates the i-th unit and ν_i is its valuation. Note that for all $i \in \{1, \cdots, \omega\}$, τ_i and ν_i are deterministic values once $\boldsymbol{\pi}$ and $\mathcal{I}$ are given.

We can derive the following inequality regarding ν_i:

$$\nu_i \geq L_i, \quad \forall i \in [\omega], \tag{A.5}$$

where L_i is the lower bound for the range of the pricing function ϕ_i, used to generate the random price for the i-th unit. This inequality holds since the buyer arriving at time τ_i accepts the price posted for the i-th unit by r-Dynamic-cost. The price for the i-th unit is at least equal to L_i based on the design of the pricing functions ϕ_i.

Let the random variable $W^{\tau_\omega}(\mathbf{P})$ denote the total number of items allocated by r-Dynamic-cost($\mathbf{P}$) after the arrival of buyer τ_ω in the instance $\mathcal{I}$. The lemma below shows that the random variable $W^{\tau_\omega}(\mathbf{P})$ is always lower bounded by $\omega - 1$.

Lemma A.1. *Given an arbitrary instance $\mathcal{I}$, $W^{\tau_\omega}(\mathbf{P}) \geq \omega - 1$ holds for all $\boldsymbol{P} \in \mathcal{P}$.*

Proof. If $\omega = 1$, this lemma is trivial, so we consider the case where $\omega \geq 2$. Suppose before the arrival of the buyer at time τ_2, no items have been sold. From Eq. (A.5), we know that $\nu_2 \geq L_2$. Additionally, based on the design of the pricing functions $\phi_1(.)$ and $\phi_2(.)$, we have $L_2 \geq U_1$. Consequently, it follows that $\nu_2 \geq U_1$. Since the realized price for the first unit under any sampled price vector will be at most U_1 (based on design of the pricing function ϕ_1), the buyer arriving at time τ_2 will accept the price for the first unit, and the algorithm will sell the first item. Thus, for all possible price vector $\mathbf{P}$, the value of the random variable $W^{\tau_2}(\mathbf{P})$ is at least equal to one. By the same reasoning, if before the arrival of the buyer at time τ_3, only one item has been sold, the buyer arriving at τ_3 will accept the price for the second unit, regardless of its price, and the total number of items sold by r-Dynamic-cost will increase to two. This reasoning can be extended to the time τ_ω. As a result, after the arrival of the buyer at time τ_ω, r-Dynamic-cost sells at least $\omega - 1$ units and thereby the claim in the lemma follows. $\square$

Lemma A.1 implies that the support of the random variable W^{τ_ω} consists only of two values: $\omega - 1$ and ω. This greatly simplifies the analysis of the algorithm.

The following two lemmas help us lower bound the expected performance of r-Dynamic-cost under the input instance $\mathcal{I}$ and upper bound the objective of the offline optimal algorithm given the instance $\mathcal{I}$, respectively.

Lemma A.2. *If a buyer in instance $\mathcal{I}$ arrives before time τ_ω with a valuation within $[L_\omega, v_{\max}]$, then for all $\mathbf{P} \in \mathcal{P}$, r-Dynamic-cost($\mathbf{P}$) will allocate one unit of the item to that buyer.*

Proof. According to the definition of $\boldsymbol{\pi}$, τ_ω is the earliest time across all possible price vectors in $\mathcal{P}$ that the production level exceeds $\omega - 1$, causing the posted price to exceed $U_{\omega-1}$. Thus, for all possible realizations of $\mathbf{P}$, the posted prices by r-Dynamic-cost remain below $U_{\omega-1}$ before the arrival of buyer at time τ_ω. Consequently, when a buyer with a valuation within $[L_\omega, v_{\max}]$ arrives before time τ_ω, the buyer accepts the price posted to him (since $L_\omega \geq U_{\omega-1}$) and a unit of item will thus be allocated to this buyer. $\square$

Lemma A.3. *There are no buyers in instance $\mathcal{I}$ with a valuation within $[U_\omega, v_{\max}]$ arriving after time τ_ω, namely, the valuations of all buyers arriving after τ_ω are less than U_ω.*

Proof. If there exists a buyer with a valuation larger than U_ω arriving after the time τ_ω, then there must exist a price vector in $\mathcal{P}$, say $\mathbf{P}'$, such that the number of units sold by r-Dynamic-cost($\mathbf{P}'$) will exceed ω. This contradicts the definition of ω. Thus, the lemma follows.[1] $\square$

Given an instance $\mathcal{I}$, let the set $\mathcal{B} \subseteq \mathcal{I}$ contain the highest-valued buyers that the offline optimal algorithm selects. We further divide $\mathcal{B}$ into two subsets: $\mathcal{B}_1$ and $\mathcal{B}_2$. $\mathcal{B}_1$ comprises the highest-valued buyers up to time τ_ω, while $\mathcal{B}_2$ includes the remaining buyers in $\mathcal{B}$ who arrive at or after time τ_ω. Let us further partition $\mathcal{B}_1$ into two subsets: $\mathcal{B}_{1,1}$ and $\mathcal{B}_{1,2}$. Here, $\mathcal{B}_{1,1}$ consists of buyers in $\mathcal{B}_1$ with valuations at least L_ω, and $\mathcal{B}_{1,2} = \mathcal{B}_1 \setminus \mathcal{B}_{1,1}$ comprises those with valuations strictly less than L_ω.

For the rest of the analysis, let us study the problem for two separate cases that may occur depending on the instance $\mathcal{I}$.

[1] In fact, such a price vector $\mathbf{P}'$ for the initial ω units should have the same prices as the vector $\boldsymbol{\pi}$ and for the $(i+1)$-th unit, $\mathbf{P}'$ should be equal to U^ω (i.e., $P'_{i+1} = U^\omega$).

Case 1: Buyer τ_ω Has the Highest Valuation

In this case, in the set $\mathcal{B}_2$, no buyer has a valuation greater than $U_{\omega-1}$ except for the buyer at time τ_ω. Therefore, the buyer at time τ_ω possesses the highest valuation in the instance $\mathcal{I}$.

Bound `OPT` from Above for Case 1. The following upper bound can be derived for $\texttt{OPT}(\mathcal{I})$, which denotes the objective value of the offline optimal algorithm on instance $\mathcal{I}$:

$$
\begin{aligned}
\texttt{OPT}(\mathcal{I}) &= V(\mathcal{B}_1) + V(\mathcal{B}_2) - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\leq V(\mathcal{B}_1) + (|\mathcal{B}_2| - 1) \cdot U_{\omega-1} + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&= V(\mathcal{B}_{1,1}) + V(\mathcal{B}_{1,2}) + (|\mathcal{B}_2| - 1) \cdot U_{\omega-1} + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\leq (k-1) \cdot U_{\omega-1} + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) + \nu_{\tau_\omega} - \sum_{i=1}^{k} c_i,
\end{aligned}
$$

where the first inequality directly follows the condition of **Case 1**. The second inequality follows the definition of $\mathcal{B}_{1,1}$ and $\mathcal{B}_{1,2}$. Finally, the third inequality follows the fact that we only focus on the case when $c_k < v_{\min}$.

Bound `ALG` from Below for Case 1. Moving forward, we focus on establishing a lower bound on the performance of r-Dynamic-cost under the arrival instance $\mathcal{I}$. Let the random variables $\{X_i\}_{\forall i \in [k]}$ represent the value obtained by r-Dynamic-cost from allocating the i-th unit of the item. Given the input instance $\mathcal{I}$, let $\mathbb{E}[\texttt{ALG}(\mathcal{I})]$ denote the expected performance of r-Dynamic-cost. Therefore, we have:

$$
\begin{aligned}
\mathbb{E}[\texttt{ALG}(\mathcal{I})] &= \mathbb{E}\left[\sum_{i=1}^{k} (X_i - c_i) \cdot \mathbf{1}_{\{i\text{-th item is sold under price vector } \mathbf{P}\}}\right], \\
&\geq \sum_{i=1}^{\omega-1} \mathbb{E}[X_i - c_i] \\
&= \sum_{i=1}^{\omega-1} \mathbb{E}[X_i] - \sum_{i=1}^{\omega-1} c_i \\
&\geq \sum_{i=1}^{\omega-1} \int_0^1 \phi_i(\eta) d\eta \ + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) - \sum_{i=1}^{\omega-1} c_i.
\end{aligned}
$$

In the equations above, all expectations are taken with respect to the randomness of the price vector **P**. The first inequality follows Lemma A.1, indicating that under any price vector **P**, r-Dynamic-cost sells at least $\omega - 1$ units. The first term in the second inequality follows due to the independent sampling used to set the price of the i-th unit using the pricing function ϕ_i, and the second term follows Lemma A.2.

Let us define $\psi_i(v) = \sup\{s : \phi_i(s) \leq v\}$ for all $i \in [k]$. From the definition of $\{\phi_i\}_{\forall i \in [k]}$ in Theorem 2.2, it follows that:

$$
\begin{aligned}
&\mathbb{E}[\texttt{ALG}(\mathcal{I})] \\
&\geq \sum_{i=1}^{\omega-1} \int_0^1 \phi_i(\eta) d\eta \; - \sum_{i=1}^{\omega-1} c_i + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) \\
&= \sum_{i=1}^{\omega} \psi_i(v_{\min}) \cdot (v_{\min} - c_{i+1}) + \sum_{i=1}^{\omega-1} \int_{\eta=v_{\min}}^{U_{\omega-1}} (\eta - c_{i+1}) d\psi_i(\eta) + \big(V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}\big).
\end{aligned}
$$

Furthermore, it is evident that based on the design of $\{\phi_i\}_{\forall i \in [k]}$ with $\alpha = \alpha^*_{\mathcal{S}}(k)$, the set of functions $\{\psi_i(v)\}$ follows the same design as $\{\psi_i^{\alpha}(v)\}$ given in Proposition 2.2. As a result, it follows that:

$$
\begin{aligned}
&\sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_{i+1}) + \sum_{i=1}^{\omega-1} \int_{\eta=v_{\min}}^{U_{\omega-1}} (\eta - c_{i+1}) d\psi_i(\eta) + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) \\
&\geq \frac{1}{\alpha^*_{\mathcal{S}}(k)} \cdot \left(k \cdot U_{\omega-1} - \sum_i c_i \right) + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) \,.
\end{aligned}
$$

Putting Everything Together for Case 1. Putting together the lower bound and upper bound derived for the expected objective value of r-Dynamic-cost and the offline optimal algorithm, it follows that:

$$
\begin{aligned}
\frac{\texttt{OPT}(\mathcal{I})}{\mathbb{E}[\texttt{ALG}(\mathcal{I})]} &\leq \frac{(k-1) \cdot U_{\omega-1} + \nu_{\tau_\omega} + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) - \sum_{i=1}^k c_i}{\frac{1}{\alpha^*_{\mathcal{S}}(k)} \cdot (k \cdot U_{\omega-1} - \sum_i c_i) + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1})} \\
&\leq \frac{(k-1) \cdot U_{\omega-1} + \nu_{\tau_\omega} - \sum_{i=1}^k c_i}{\frac{1}{\alpha^*_{\mathcal{S}}(k)} \cdot (k \cdot U_{\omega-1} - \sum_i c_i)} \\
&= \alpha^*_{\mathcal{S}}(k) \cdot \left(1 + \frac{\nu_{\tau_\omega} - U_{\omega-1}}{k \cdot U_{\omega-1} - C} \right) \\
&\leq \alpha^*_{\mathcal{S}}(k) \cdot \left(1 + \frac{U_\omega - U_{\omega-1}}{k \cdot U_{\omega-1} - C} \right) \\
&\leq \alpha^*_{\mathcal{S}}(k) \cdot e^{\frac{\alpha^*_{\mathcal{S}}(k)}{k}} .
\end{aligned}
$$

In the equation above, the last inequality is due to the fact that $\frac{U_\omega - U_{\omega-1}}{U_{\omega-1}-c_\omega} = \frac{U_\omega - c_\omega}{U_{\omega-1}-c_\omega} - 1 \le 1 + e^{\frac{\alpha^*_{\mathcal{S}}(k)}{k}}$, where the last inequality follows the design in Eq. (2.10).

Case 2: Buyer τ_ω Does Not Have the Highest Valuation

In the set of buyers $\mathcal{B}_2$, there are other buyers with valuation greater than $U_{\omega-1}$ besides the buyer at time τ_ω. Let λ denote the value of the highest buyer in $\mathcal{B}_2$ along with the value of buyer at time τ_ω. First, let us consider the case that $\lambda \le \nu_{\tau_\omega}$. The proof for the case that $\lambda > \nu_{\tau_\omega}$ follows exactly the same as the following case.

Bound `OPT` from Above for Case 2. Following the same approach as the previous Case 1, let us first upper bound the objective of the offline optimal algorithm on instance $\mathcal{I}$:

$$\begin{aligned}
\texttt{OPT}(\mathcal{I}) &= V(\mathcal{B}_1) + V(\mathcal{B}_2) - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\le V(\mathcal{B}_1) + (|\mathcal{B}_2| - 1)\cdot\lambda + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\le V(\mathcal{B}_{1,1}) + V(\mathcal{B}_{1,2}) + (|\mathcal{B}_2| - 1)\cdot\lambda + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\le (k-1)\cdot\lambda + \nu_{\tau_\omega} + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}|\cdot U_{\omega-1}) - \sum_{i=1}^{k} c_i.
\end{aligned}$$

Bound `ALG` from Below for Case 2. To establish a lower bound on the performance of r-Dynamic-cost in this case, let us consider the following lemma:

Lemma A.4. *If the random price of the ω-th unit is realized to be less than λ and further assume that $\lambda \le \nu_{\tau_\omega}$, then the number of items allocated by r-Dynamic-cost in the end is equal to ω.*

Proof. Under any price realization, as established by Lemma A.1, it is proven that after the arrival of the buyer at time τ_ω, the number of allocated units is at least $\omega - 1$. If the price of the ω-th unit is realized to be less than λ, then upon the arrival of the buyer with valuation λ at some time after τ_ω, the buyer will accept the price if the ω-th unit has not already been sold. □

Next, we obtain a lower bound on the performance of r-Dynamic-cost as follows:

$$
\begin{aligned}
&\mathbb{E}[\texttt{ALG}(\mathcal{I})] \\
&= \mathbb{E}\left[\sum_{i=1}^{k}(X_i - c_i)\cdot \mathbf{1}_{\{i\text{-th item is sold under pricie vector } \mathbf{P}\}}\right] \\
&\geq \sum_{i=1}^{\omega-1}\mathbb{E}[X_i - c_i] + \mathbb{E}[X_\omega - c_\omega | P_\omega \leq \lambda] \\
&\geq \sum_{i=1}^{\omega-1}\int_0^1 \phi_i(\eta)d\eta \; + \int_0^{\phi_\omega^{-1}(\lambda)} \phi_\omega(\eta)d\eta - \phi_\omega^{-1}(\lambda)\cdot c_\omega - \sum_{i=1}^{\omega-1} c_i + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}).
\end{aligned}
$$

In the equations above, all expectations are taken with respect to the randomness of the price vector $\mathbf{P} \in \mathcal{P}$. The first inequality follows Lemma A.4, where P_ω denotes the ω-th element of the random price vector $\mathbf{P}$ that r-Dynamic-cost posts for the ω-th unit. The second inequality is true because of the independent sampling that is used to set the random price of the i-th unit using ϕ_i and Lemma A.2.

Let us define $\psi_i(v) = \sup\{s : \phi_i(s) \leq v\}$, $i \in [k]$. From the definition of $\{\phi_i\}_{i\in[k]}$ in Theorem 2.2, it follows that:

$$
\begin{aligned}
&\sum_{i=1}^{\omega-1}\int_0^1 \phi_i(\eta)d\eta \; + \int_0^{\phi_\omega^{-1}(\lambda)} \phi_\omega(\eta)d\eta - \phi_\omega^{-1}(\lambda)\cdot c_\omega - \sum_{i=1}^{\omega-1} c_i + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}) \\
&= \sum_{i=1}^{k}\psi_i(v_{\min})\cdot(v_{\min} - c_i) + \sum_{i=1}^{\omega}\int_{\eta=v_{\min}}^{\lambda}(\eta - c_i)d\psi_i(\eta) + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}).
\end{aligned}
$$

Furthermore, it is evident that the set of functions $\{\psi_i(v)\}_{\forall i\in[k]}$ follows the same design as $\{\psi_i^\alpha(v)\}_{i\in[k]}$ given in Lemma 2.2 (recall that $\{\psi_i^\alpha(v)\}_{i\in[k]}$ are based on $\{\phi_i\}_{\forall i\in[k]}$). As a result, it follows that:

$$
\begin{aligned}
&\sum_{i=1}^{k}\psi_i(v_{\min})\cdot(v_{\min} - c_i) + \sum_{i=1}^{\omega}\int_{\eta=v_{\min}}^{\lambda}(\eta - c_i)d\psi_i(\eta) + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}) \\
&\geq \frac{1}{\alpha_{\mathcal{S}}^*(k)}\cdot\left(k\cdot\lambda - \sum_i c_i\right) + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}).
\end{aligned}
$$

Putting Everything Together for Case 2. Putting together the above lower and upper bounds, it follows that:

$$\begin{aligned}\frac{\texttt{OPT}(\mathcal{I})}{\mathbb{E}[\texttt{ALG}(\mathcal{I})]} &\le \frac{(k-1)\cdot\lambda + \nu_{\tau_\omega} + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1}) - \sum_{i=1}^k c_i}{\frac{1}{\alpha^*_{\mathcal{S}}(k)}\cdot\left(k\cdot\lambda - \sum_{i=1}^k c_i\right) + (V(\mathcal{B}_1) - |\mathcal{B}_1|\cdot U_{\omega-1})} \\ &\le \frac{(k-1)\cdot\lambda + \nu_{\tau_\omega} - \sum_{i=1}^k c_i}{\frac{1}{\alpha^*_{\mathcal{S}}(k)}\cdot\left(k\cdot\lambda - \sum_{i=1}^k c_i\right)} \\ &= \alpha^*_{\mathcal{S}}(k)\cdot\left(1 + \frac{\nu_{\tau_\omega} - \lambda}{k\cdot\lambda - C}\right) \\ &\le \alpha^*_{\mathcal{S}}(k)\cdot\left(1 + \frac{U_\omega - U_{\omega-1}}{k\cdot U_{\omega-1} - C}\right) \\ &\le \alpha^*_{\mathcal{S}}(k)\cdot e^{\frac{\alpha^*_{\mathcal{S}}(k)}{k}}.\end{aligned}$$

We thus complete the proof of Theorem 2.2.

A.7 Proof of Corollary 2.1

In this subsection, we prove that for an arbitrary instance $\mathcal{I}$, the expected performance of R-DYNAMIC-COST, denoted as $\mathbb{E}[\texttt{ALG}(\mathcal{I})]$, is at least $\frac{\texttt{OPT}(\mathcal{I})}{\alpha^*_{\mathcal{S}}(2)}$.

Let v_1^*, v_2^* denote the two highest valuations in the instance $\mathcal{I}$ (we omit the proof for the trivial case with only one buyer in $\mathcal{I}$). Depending on the values of v_1^* and v_2^*, the following three cases occur. In each scenario, we prove that $\mathbb{E}[\texttt{ALG}(\mathcal{I})] \ge \frac{\texttt{OPT}(\mathcal{I})}{\alpha^*_{\mathcal{S}}(2)} = \frac{v_1^*+v_2^*-c_1-c_2}{\alpha^*_{\mathcal{S}}(2)}$ holds.

Case I: $v_1^* \le v_2^* \le U_1$. Let random variables X_1 and X_2 denote the valuations of the buyers that purchase the first and second unit of the item, respectively. Then, it follows that

$$\begin{aligned}\mathbb{E}[\texttt{ALG}(\mathcal{I})] &= \mathbb{E}_{\boldsymbol{s}\sim U^2(0,1)}[X_1 + X_2 - c_1 - c_2] \\ &\ge \int_{s_1=0}^{\phi_1^{-1}(v_1^*)} (\phi_1(s_1) - c_1)\cdot ds_1 + (v_2^* - c_1)\cdot(\phi_1^{-1}(v_2^*) - \phi_1^{-1}(v_1^*)), \\ &\ge \int_{s_1=0}^{\phi_1^{-1}(v_1^*)} (\phi_1(s_1) - c_1)\cdot ds_1 + \int_{s_1=\phi_1^{-1}(v_1^*)}^{\phi_1^{-1}(v_2^*)} (\phi_1(s_1) - c_1)\cdot ds_1 \\ &= \int_{s_1=0}^{\phi_1^{-1}(v_2^*)} (\phi_1(s_1) - c_1)\cdot ds_1 \\ &\ge \frac{v_1^* + v_2^* - c_1 - c_2}{\alpha^*_{\mathcal{S}}(2)} \\ &= \frac{\texttt{OPT}(\mathcal{I})}{\alpha^*_{\mathcal{S}}(2)},\end{aligned}$$

where the first two terms in the first inequality arise from the fact that if the realized price for the first unit of the item, denoted as $P_1 = \phi_1(s_1)$, is set below v_1^*, then in the worst-case scenario, the value obtained from the first item will be at least equal to $\phi_1(s_1)$. The subsequent two terms are included because if the price for the first item falls within the range from v_1^* to v_2^*, then the first item is allocated to the buyer whose valuation is v_2^*. The second inequality follows since $\phi_1(s_1)$ is a non-decreasing function. The third inequality follows from the design of $\phi_1(s_1)$ in Theorem 2.1.

Case II: $v_1^* \leq U_1 = L_2 \leq v_2^* \leq v_{\max}$**.** In this case, we have

$$
\begin{aligned}
&\mathbb{E}[\texttt{ALG}(\mathcal{I})] \\
=& \mathbb{E}_{\boldsymbol{s}\sim U^2(0,1)}[X_1 - c_1] + \mathbb{E}_{\boldsymbol{s}\sim U^2(0,1)}[X_2 - c_2] \\
\geq& \int_{s_1=0}^{\phi_1^{-1}(v_1^*)} (\phi_1(s_1) - c_1) ds_1 + (v_2^* - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) + (v_2^* - c_2) \cdot \phi_1^{-1}(v_1^*) \cdot \phi_2^{-1}(v_2^*) \\
=& \frac{2 \cdot v_1^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) + (v_2^* - c_2) \cdot \phi_1^{-1}(v_1^*) \cdot \phi_2^{-1}(v_2^*).
\end{aligned}
$$

To prove $\mathbb{E}[\texttt{ALG}(\mathcal{I})] \geq \frac{\texttt{OPT}(\mathcal{I})}{\alpha_{\mathcal{S}}^*(2)} = \frac{v_1^* + v_2^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)}$, we define the following function

$$
\begin{aligned}
G(v_1^*, v_2^*) = & \frac{2 \cdot v_1^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) + \\
& (v_2^* - c_2) \cdot \phi_1^{-1}(v_1^*) \cdot \phi_2^{-1}(v_2^*) - \frac{v_1^* + v_2^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)}.
\end{aligned}
$$

Then the goal is to prove $G(v_1^*, v_2^*) \geq 0$ in its domain $v_{\min} \leq v_1^* \leq U_1$ and $L_2 \leq v_2^* \leq v_{\max}$. The proposition below formally states this result.

Proposition A.1. *For all $v_1^* \in [L_1, U_1]$ and $v_2^* \in [L_2, U_2]$, we have $G(v_1^*, v_2^*) \geq 0$.*

We deferred the proof of the above proposition to Appendix A.8. The idea is to simply prove that $G(v_1^*, v_2^*) \geq 0$ holds at all extreme points within its domain.

Case III: $L_2 \leq v_1^* \leq v_2^*$**.** In this case, we show that we can lower bound the expected performance of r-Dynamic-cost as follows:

$$
\begin{aligned}
&\mathbb{E}[\texttt{ALG}(\mathcal{I})] \\
=&\, \mathbb{E}_{\boldsymbol{s}\sim U^2(0,1)}[X_1 - c_1] + \mathbb{E}_{\boldsymbol{s}\sim U^2(0,1)}[X_2 - c_2] \\
\geq& \int_{s_1=0}^{1} (\phi_1(s_1) - c_1)\cdot ds_1 + \int_{s_2=0}^{\phi_2^{-1}(v_1^*)} (\phi_2(s_2) - c_2)\cdot ds_2 + (v_2^* - c_2)\cdot(\phi_2^{-1}(v_2^*) - \phi_2^{-1}(v_1^*)) \\
\geq& \int_{s_1=0}^{1} (\phi_1(s_1) - c_1)\cdot ds_1 + \int_{s_2=0}^{\phi_2^{-1}(v_1^*)} (\phi_2(s_2) - c_2)\cdot ds_2 + \int_{s_2=\phi_2^{-1}(v_1^*)}^{\phi_2^{-1}(v_2^*)} (\phi_2(s_2) - c_2)\cdot ds_2 \\
=& \int_{s_1=0}^{1} (\phi_1(s_1) - c_1)\cdot ds_1 + \int_{s_2=0}^{\phi_2^{-1}(v_2^*)} (\phi_2(s_2) - c_2)\cdot ds_2 \\
=& \frac{2\cdot U_1 - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + \int_{s_2=0}^{\phi_2^{-1}(v_2^*)} (\phi_2(s_2) - c_2)\cdot ds_2 \\
=& \frac{2\cdot v_2^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& \frac{\texttt{OPT}(\mathcal{I})}{\alpha_{\mathcal{S}}^*(2)},
\end{aligned}
$$

where the first term in the first inequality arises from the fact that if the realized price for the first unit of the item, denoted as $P_1 = \phi_1(s_1)$, is set below L_2, then in the worst-case scenario, the value obtained from the first item will be at least equal to $\phi_1(s_1)$. The second and third terms follow the same reasoning. The second inequality follows the fact that $\phi_2(s_2)$ is non-decreasing. The third and forth equalities follow the design of $\phi_1(s_1)$ and $\phi_2(s_2)$ in Theorem 2.1.

Combining the analysis of the above three cases, Corollary 2.1 follows.

A.8 Proof of Proposition A.1

We first evaluate the value of $G(v_1^*, v_2^*)$ at its critical points, that is, at the points where $\frac{\partial G(v_1^*,v_2^*)}{\partial v_1^*} = 0$ and $\frac{\partial G(v_1^*,v_2^*)}{\partial v_2^*} = 0$, and show that $G(v_1^*, v_2^*) \geq 0$ holds at these critical points. After that, the proposition follows by evaluating the values of $G(v_1^*, v_2^*)$ at the four boundary hyperplanes of its domain.

First, let us compute $\frac{\partial G(v_1^*,v_2^*)}{\partial v_1^*}$. It follows that:

$$\frac{\partial G(v_1^*, v_2^*)}{\partial v_1^*} = \frac{1}{\alpha_{\mathcal{S}}^*(2)} - \frac{2}{\alpha_{\mathcal{S}}^*(2)}\cdot\frac{v_2^* - c_1}{v_1^* - c_1} + \frac{2}{\alpha_{\mathcal{S}}^*(2)}\cdot\frac{v_2^* - c_2}{v_1^* - c_1}\cdot\phi_2^{-1}(v_2^*).$$

Setting the right-hand side of the above equation to zero, we have

$$\phi_2^{-1}(v_2^*) \cdot (v_2^* - c_2) = v_2^* - c_1 - \frac{v_1^* - c_1}{2}.$$

Using the equation above, we then compute $G(v_1^*, v_2^*)$ at the points that $\frac{\partial G(v_1^*, v_2^*)}{\partial v_1^*} = 0$, it follows that:

$$\begin{aligned}
G(v_1^*, v_2^*) =& \frac{v_1^* - v_2^*}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) + (v_2^* - c_1 - \frac{v_1^* - c_1}{2}) \cdot \phi_1^{-1}(v_1^*) \\
=& \frac{v_1^* - v_2^*}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_1) - \frac{v_1^* - c_1}{2} \cdot \phi_1^{-1}(v_1^*) \\
\geq& \frac{v_1^* - v_2^*}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_1) - \frac{v_1^* - c_1}{2} \\
=& v_1^* \cdot \left(\frac{1}{\alpha_{\mathcal{S}}^*(2)} - \frac{1}{2} \right) + v_2^* \cdot \left(1 - \frac{1}{\alpha_{\mathcal{S}}^*(2)} \right) - \frac{c_1}{2} \\
\geq& \frac{v_1^* - c_1}{2} \\
>& 0,
\end{aligned}$$

leading to the conclusion that $G(v_1^*, v_2^*) \geq 0$ holds at its critical points.

Next, we consider the boundary hyperplanes and prove that $G(v_1^*, v_2^*)$ is positive in all four boundary planes given below:

- $G(L_1, v_2^*), \quad \forall v_2^* \in [L_2, U_2]$.
- $G(U_1, v_2^*), \quad \forall v_2^* \in [L_2, U_2]$.
- $G(v_1^*, L_2), \quad \forall v_1^* \in [L_1, U_1]$.
- $G(v_1^*, U_2), \quad \forall v_1^* \in [L_1, U_1]$.

We start with the first one $G(L_1, v_2^*)$:

$$\begin{aligned}
G(L, v_2^*) =& \frac{2 \cdot L - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_2) - \frac{L + v_2^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} \\
=& (v_2^* - c_2) - \frac{v_2^* - L}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& 0, \qquad \forall L_2 \leq v_2^* \leq U_2,
\end{aligned}$$

where the equations above follow since $L \geq c_2$ holds (the assumption that the marginal production costs are always less than the valuations).

For the second one $G(U_1, v_2^*)$:

$$
\begin{aligned}
G(U_1, v_2^*) =& \frac{2 \cdot U_1 - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (v_2^* - c_2) \cdot \phi_2^{-1}(v_2^*) - \frac{U_1 + v_2^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} \\
=& (v_2^* - c_2) \cdot \phi_2^{-1}(v_2^*) - \frac{v_2^* - U_1}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& 0, \qquad \forall L_2 \leq v_2^* \leq U_2 = v_{\max}.
\end{aligned}
$$

The equations above follow since $(v_2^* - c_2) \cdot \phi_2^{-1}(v_2^*) \geq \int_{s_2=0}^{\phi_2^{-1}(v_2^*)} (\phi_2(s_2) - c_2) \cdot ds_2 \geq 2 \cdot \frac{v_2^* - U_1}{\alpha_{\mathcal{S}}^*(2)}$ based on the definition of $\phi_2(s)$.

For the third one $G(v_1^*, L_2)$:

$$
\begin{aligned}
G(v_1^*, L_2) =& \frac{2 \cdot v_1^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (L_2 - c_1) \cdot \phi_1^{-1}(v_1^*) - \frac{v_1^* + L_2 - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} \\
=& (L_2 - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) - \frac{L_2 - v_1^*}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& 0, \qquad \forall L_1 \leq v_1^* \leq U_1,
\end{aligned}
$$

where the above equation follows since $(L_2 - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) \geq \int_{s_1=\phi_1^{-1}(v_1^*)}^{\phi_1^{-1}(L_2)} (\phi_1(s_1) - c_1) \cdot ds_1 \geq 2 \cdot \frac{L_2 - v_1^*}{\alpha_{\mathcal{S}}^*(2)}$ (based on the definition of $\phi_1(s)$).

Finally, for the last one $G(v_1^*, U_2)$:

$$
\begin{aligned}
G(v_1^*, U_2) =& \frac{2 \cdot v_1^* - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} + (U_2 - c_1) \cdot (1 - \phi_1^{-1}(v_1^*)) + (U_2 - c_2) \cdot \phi_1^{-1}(v_1^*) - \frac{v_1^* + U_2 - c_1 - c_2}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& (U_2 - c_2) - \frac{U_2 - v_1^*}{\alpha_{\mathcal{S}}^*(2)} \\
\geq& 0, \qquad \forall L = L_1 \leq v_1^* \leq U_1,
\end{aligned}
$$

where the equations above follow since $v_1^* \geq c_2$ holds (again, the assumption that the marginal production costs are always less than the valuations).

Combining all the above analysis, we thus complete the proof of Proposition A.1.

A.9 Extension of the Lower Bound Results to General Production Cost Functions

In this subsection, we extend our lower bound result in Theorem 2.1, originally developed for the high-value case,[2] to general cumulative production cost functions.

Before presenting the main theorem on obtaining a lower bound for general cost functions, let us introduce some notations. Define $f^*(v) : [v_{\min}, v_{\max}] \to \mathbb{R}$ as the conjugate of the total production cost function, where $f^*(v) = \max_{i \in [k]} \big(v \cdot i - f(i)\big)$. Additionally, let $g(v)$ be defined as

$$g(v) = (f^*)'(v) = \sum_{i \in [k]} \mathbf{1}_{\{v \geq c_i\}},$$

where $\mathbf{1}_{\{A\}}$ is the standard indicator function. Let $\underline{k}$ denote the smallest natural number such that:

$$\sum_{i=1}^{\underline{k}} (v_{\min} - c_i) > \frac{1}{\alpha} \cdot f^*(v_{\min}).$$

Following Theorem 2.1, we also define ξ as follows:

$$\xi = \frac{\frac{1}{\alpha} \cdot f^*(v_{\min}) - \sum_{i=1}^{\underline{k}-1} (v_{\min} - c_i)}{v_{\min} - c_{\underline{k}}}.$$

Theorem A.1 below extends our lower bound results to settings with general cost functions.

Theorem A.1. *Given $\mathcal{S} = \{v_{\min}, v_{\max}, f\}$ for the* `kSelection-cost` *problem with $k \geq 1$ and general production cost functions f, no online algorithm, including those with randomization, can achieve a competitive ratio smaller than $\alpha^*_{\mathcal{S}}(k)$, where $\alpha^*_{\mathcal{S}}(k)$ is the solution to the following system of equations of α:*

$$\int_{\eta = v_{\min}}^{u_{\underline{k}}} \frac{g(\eta)}{\alpha \cdot (\eta - c_{\underline{k}})} d\eta = 1 - \xi, \tag{A.6}$$

$$\int_{\eta = \ell_i}^{u_i} \frac{g(\eta)}{\alpha \cdot (\eta - c_i)} d\eta = 1, \quad u_i = \ell_{i+1}, \quad i = \underline{k} + 1, \ldots, k, \tag{A.7}$$

$$u_k = v_{\max}. \tag{A.8}$$

Proof. The proof proceeds similarly to the proof of Theorem 2.1 until the derivation of Eq. (2.5). Given the arrival instance $\mathcal{I}^{(\epsilon)}$ up to the end of stage-v, the objective of the offline optimal algorithm

[2] This corresponds to the case when $c_k < v_{\min}$, or equivalently, the lowest possible valuation $v_{\min}$ is no less than the highest marginal production cost c_k.

equals $f^*(v)$. Therefore, we reformulate Eq. (2.4) as follows:

$$\texttt{ALG}\left(\mathcal{I}_v^{(\epsilon)}\right) \geq \frac{1}{\alpha} \cdot f^*(v), \quad \forall v \in [v_{\min}, v_{\max}].$$

In the case of general production cost functions, we derive the following inequality to capture the production level changes of an α-competitive algorithm:

$$\sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \sum_{i=1}^{k} \int_{\eta=v_{\min}}^{v} (\eta - c_i) d\psi_i(\eta) \geq \frac{1}{\alpha} \cdot f^*(v). \tag{A.9}$$

In addition, we define $\alpha^*_{\mathcal{S}}(k)$ as follows:

$$\alpha^*_{\mathcal{S}}(k) = \inf \Big\{\alpha \geq 1 \Big| \text{there exist a set of } k \text{ allocation functions } \{\psi_i(v)\}_{\forall i \in [k]} \in \Omega \text{ that satisfy Eq. (A.9)}\Big\}.$$

From this point onward, the proof continues in the same manner as the proof of Theorem 2.1. Let us define the function $\chi^\alpha(v) : [v_{\min}, v_{\max}] \to [0, k]$ and the set of functions $\{\psi_i^\alpha(v)\}_{i \in [k]}$ as specified in Eq. (2.6) and Eq. (2.2). Consequently, Lemma 2.3 holds as long as we have increasing marginal production costs (i.e., diseconomies of scale) and Lemma 2.4 that follows the definition of $\{\psi_i^\alpha(v)\}_{i \in [k]}$ holds in this case as well.

The primary distinction between the two proofs arises in the following proposition, which gives an explicit design of the function $\{\psi_i^\alpha\}_{\forall i \in [k]}$ by replacing the inequality with an equality in Eq. (A.9).

Proposition A.2. *For any $\alpha \geq \alpha^*_{\mathcal{S}}(k)$, there exist a unique set of functions $\{\psi_i^\alpha(v)\}_{\forall i \in [k]}$ that satisfy Eq.* (A.9) *with an equality:*

$$\psi_i^\alpha(v) = 1, \quad \forall v \in [v_{\min}, v_{\max}], \quad 1 \leq i \leq \underline{k} - 1,$$

$$\psi_{\underline{k}}^\alpha(v) = \begin{cases} 0 & v \leq \ell_{\underline{k}}, \\ \xi + \int_{\eta=v_{\min}}^{v} \frac{g(\eta)}{\alpha \cdot (\eta - c_i)} d\eta, & v \in [v_{\min}, u_{\underline{k}}], \\ 1 & v \geq u_{\underline{k}}, \end{cases}$$

$$\psi_i^\alpha(v) = \begin{cases} 0 & v \leq \ell_i, \\ \int_{\eta=\ell_i}^{v} \frac{g(\eta)}{\alpha \cdot (\eta - c_i)} d\eta, & v \in [\ell_i, u_i], \\ 1 & v \geq u_i, \end{cases} \quad i = \underline{k} + 1, \ldots, k - 1,$$

$$\psi_k^\alpha(v) = \begin{cases} 0 & v \leq \ell_k, \\ \int_{\eta=\ell_k}^{v} \frac{g(\eta)}{\alpha \cdot (\eta - c_k)} d\eta, & v \in [\ell_k, v_{\max}], \end{cases}$$

where the intervals are specified by:

$$\int_{\eta=v_{\min}}^{u_{\underline{k}}} \frac{g(\eta)}{\alpha\cdot(\eta-c_{\underline{k}})}d\eta = 1-\xi, \tag{A.10}$$

$$\int_{\eta=\ell_i}^{u_i} \frac{g(\eta)}{\alpha\cdot(\eta-c_i)}d\eta = 1, \quad u_i=\ell_{i+1}, \quad \forall i=\underline{k}+1,\dots,k. \tag{A.11}$$

In the proposition above, for any given $\alpha \geq \alpha^*_{\mathcal{S}}(k)$, the values of u_i and ℓ_i can be determined. We begin by solving Eq. (A.10) to find the value of $u_{\underline{k}}$, and then proceed to find the value of other variables $\{u_i\}_{\forall i}$ using Eq. (A.11).

Based on the above proposition, as the value of α decreases, the value of u_k also decreases. Again, following the same reasoning as the proof of Theorem 2.1, the lower bound $\alpha^*_{\mathcal{S}}(k)$ is the value of α for which u_k computed above is equal to $v_{\max}$. We thus complete the proof of Theorem A.1. □

A.10 Extension of the Upper Bound Results to General Production Cost Functions

In this subsection, we extend the randomized dynamic pricing scheme R-DYNAMIC-COST, originally developed for the high-value case, to general cumulative production cost functions.

Theorem A.2. *Given $\mathcal{S} = \{v_{\min}, v_{\max}, f\}$ for the `kSelection-cost` problem with $k \geq 1$, R-DYNAMIC-COST (Algorithm 1) is $\max_{i\in[k]} \alpha^*_{\mathcal{S}}(k)\cdot\left(1+\frac{U_i-c_i}{f^*(U_{i-1})}\right)$-competitive for the following design of the pricing functions $\{\phi_i\}_{\forall i\in[k]}$, where $\alpha^*_{\mathcal{S}}(k)$ is the lower bound obtained in Theorem A.1:*

$$\phi_i(s) = v_{\min}, \quad \forall s\in[0,1], \quad i\in[\underline{k}^*-1],$$

$$\phi_{\underline{k}^*}(s) = \begin{cases} v_{\min} & s\in[0,\xi^*], \\ \psi^{-1}_{\underline{k}^*}(s) & s\in(\xi^*,1], \end{cases}$$

$$\phi_i(s) = \psi_i^{-1}(s), \quad \forall s\in[0,1], \quad i=\underline{k}^*+1,\dots,k,$$

where the set of functions $\{\psi_i(v)\}_{\forall i\in\{\underline{k}^*,\cdots,k\}}$ are defined as follows:

$$\psi_{\underline{k}^*}(v) = \xi^* + \int_{\eta=v_{\min}}^{v} \frac{g(\eta)}{\alpha\cdot(\eta-c_i)}d\eta, \quad \forall v\in[v_{\min}, U_{\underline{k}^*}],$$

$$\psi_i(v) = \int_{\eta=\ell_i}^{v} \frac{g(\eta)}{\alpha\cdot(\eta-c_i)}d\eta, \quad \forall v\in[L_i,U_i], \;\; i=\underline{k}^*+1,\dots,k;$$

the parameters $\underline{k}^*$ and ξ^* are respectively the values of $\underline{k}$ and ξ defined in Appendix A.9, correspond-

ing to $\alpha = \alpha^*_{\mathcal{S}}(k)$, and the price intervals $\{[L_i, U_i]\}_{\forall i\in[k]}$ are given as follows:

$$\int_{\eta=v_{\min}}^{U_{\underline{k}^*}} \frac{g(\eta)}{\alpha\cdot(\eta - c_{\underline{k}})} d\eta = 1-\xi,$$
$$\int_{\eta=L_i}^{U_i} \frac{g(\eta)}{\alpha\cdot(\eta - c_i)} d\eta = 1, \quad u_i = \ell_{i+1}, \quad \forall i = \underline{k}^*+1,\dots,k.$$

Proof. The proof will follow the same process as the proof in Appendix A.6. So we only provide a brief proof sketch.

Consider an arbitrary arrival instance $\mathcal{I} = \{v_n\}_{n\in[N]}$. Recall that the random price vector $\mathbf{P} = \{P_1,\cdots,P_k\}$ is generated using the pricing functions $\{\phi_i\}_{\forall i\in[k]}$ at the beginning of R-DYNAMIC-COST. Let us define the random variable $W(\mathbf{P})$, the variable ω and the price vector $\boldsymbol{\pi}$, the set $\{\nu_i,\tau_i\}_{\forall i\in[\omega]}$, and $W^{\tau_\omega}(\mathbf{P})$ in the same fashion as in Appendix A.6.

Following the same reasoning, the property in Eq. (A.5) can be derived for $\{\nu_i\}_{\forall i\in[\omega]}$, and the lemmas A.1, A.2, and A.3 follow as well.

We also define $\mathcal{B}\subseteq\mathcal{I}$, as before, to be the set of highest-valued buyers to whom the offline optimal algorithm allocates a unit of the item in instance $\mathcal{I}$. We further divide $\mathcal{B}$ into two subsets: $\mathcal{B}_1$ and $\mathcal{B}_2$, as done in the previous proof. Additionally, we partition $\mathcal{B}_1$ into two subsets: $\mathcal{B}_{1,1}$ and $\mathcal{B}_{1,2}$, as before.

We continue our analysis for two separate cases that can arise depending on the instance $\mathcal{I}$. In this proof, we only provide the proof for the first case and the proof of the second case follows similarly as Appendix A.6.

Case 1: In this case, no buyer in $\mathcal{B}_2$ has a valuation greater than $U_{\omega-1}$ except for the buyer at time τ_ω. Therefore, the buyer at time τ_ω possesses the highest valuation in instance $\mathcal{I}$. The following upper bound can be derived for $\texttt{OPT}(\mathcal{I})$, which denotes the objective value of the offline optimal algorithm:

$$\begin{aligned}
\texttt{OPT}(\mathcal{I}) &= V(\mathcal{B}_1) + V(\mathcal{B}_2) - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\le V(\mathcal{B}_1) + (|\mathcal{B}_2|-1)\cdot U_{\omega-1} + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&= V(\mathcal{B}_{1,1}) + V(\mathcal{B}_{1,2}) + (|\mathcal{B}_2|-1)\cdot U_{\omega-1} + \nu_{\tau_\omega} - \sum_{i=1}^{|\mathcal{B}|} c_i \\
&\le (k-1)\cdot U_{\omega-1} + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}|\cdot U_{\omega-1}) + \nu_{\tau_\omega} - \sum_{i=1}^{k} c_i,
\end{aligned}$$

where the first inequality follows the condition of **Case 1**. The second inequality follows the definition of the sets $\mathcal{B}_{1,1}$ and $\mathcal{B}_{1,2}$. Finally, the third inequality follows since based on definition of f^*, we have $(|\mathcal{B}_{1,1}| + |\mathcal{B}_{1,2}| + |\mathcal{B}_2| - 1) \cdot U_{\omega-1} - \sum_{i=1}^{|\mathcal{B}|-1} c_i \leq f^*(U_{\omega-1})$.

Moving forward, we can lower bound the expected performance of r-Dynamic-cost under $\mathcal{I}$, denoted by $\mathbb{E}[\texttt{ALG}(\mathcal{I})]$, using the same approach as before.

$$\mathbb{E}[\texttt{ALG}(\mathcal{I})] \geq \sum_{i=1}^{\omega-1} \int_0^1 \phi_i(\eta) d\eta \ + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) - \sum_{i=1}^{\omega-1} c_i.$$

Based on the definition of $\{\phi_i\}_{\forall i \in [k]}$, we have:

$$\begin{aligned}\mathbb{E}[\texttt{ALG}(\mathcal{I})] &\geq \sum_{i=1}^{\omega-1} \int_0^1 \phi_i(\eta) d\eta \ - \sum_{i=1}^{\omega-1} c_i + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) \\ &= \sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \sum_{i=1}^{\omega-1} \int_{\eta=v_{\min}}^{U_{\omega-1}} (\eta - c_i) d\psi_i(\eta) + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}).\end{aligned}$$

Furthermore, based on the design of $\{\psi_i(v)\}_{\forall i \in [k]}$ in Theorem A.2, we have

$$\begin{aligned}&\sum_{i=1}^{k} \psi_i(v_{\min}) \cdot (v_{\min} - c_i) + \sum_{i=1}^{\omega-1} \int_{\eta=v_{\min}}^{U_{\omega-1}} (\eta - c_i) d\psi_i(\eta) + (V(\mathcal{B}_1) - |\mathcal{B}_1| \cdot U_{\omega-1}) \\ \geq\ & \frac{1}{\alpha^*_{\mathcal{S}}(k)} f^*(U_{\omega-1}) + (V(\mathcal{B}_1) - |\mathcal{B}_1| \cdot U_{\omega-1}).\end{aligned}$$

Putting together the above lower and upper bounds, it follows that:

$$\begin{aligned}\frac{\texttt{OPT}(\mathcal{I})}{\mathbb{E}[\texttt{ALG}(\mathcal{I})]} &\leq \frac{f^*(U_{\omega-1}) + (V(\mathcal{B}_{1,1}) - |\mathcal{B}_{1,1}| \cdot U_{\omega-1}) + \nu_{\tau_\omega} - c_\omega}{\frac{1}{\alpha^*_{\mathcal{S}}(k)} f^*(U_{\omega-1}) + (V(\mathcal{B}_1) - |\mathcal{B}_1| \cdot U_{\omega-1})} \\ &\leq \frac{f^*(U_{\omega-1}) + \nu_{\tau_\omega} - c_\omega}{\frac{1}{\alpha^*_{\mathcal{S}}(k)} f^*(U_{\omega-1})} \\ &= \alpha^*_{\mathcal{S}}(k) \cdot \left(1 + \frac{\nu_{\tau_\omega} - c_\omega}{f^*(U_{\omega-1})}\right) \\ &\leq \alpha^*_{\mathcal{S}}(k) \cdot \left(1 + \frac{U_\omega - c_\omega}{f^*(U_{\omega-1})}\right) \\ &\leq \max_{i \in [k]} \alpha^*_{\mathcal{S}}(k) \cdot \left(1 + \frac{U_i - c_i}{f^*(U_{i-1})}\right).\end{aligned}$$

Case 2: In the set of buyers $\mathcal{B}_2$, there are other buyers with valuation greater than $U_{\omega-1}$ besides the buyer at time τ_ω. The proof in this case follows the same structure as the proof above and the proof in Appendix A.6. □

Appendix B

Proofs of Chapter 3

B.1 Proof of Theorem 3.2

We use an online primal-dual approach to establish the competitive ratio of Algorithm 2. Consider the dual LP corresponding to the primal LP in Eq. (3.1):

$$\min_{\lambda,\mathbf{u}} \quad \sum_{n\in[N]} u_n + k \cdot \sum_{n\in[N]} \lambda_n, \tag{B.1}$$

$$\text{s.t.} \quad u_n + \sum_{j=n}^{N} \lambda_j \mathbf{1}_{\{a_n+d>a_j\}} \geq v_n, \quad \forall n \in [N], \tag{B.2}$$

$$\lambda_n \geq 0, \quad \forall n \in [N]. \tag{B.3}$$

Let $\texttt{ALG}(\mathcal{I})$ represent the expected objective value of Algorithm 2 on an instance $\mathcal{I}$. Moving forward, we first propose a feasible solution for the primal problem given in Eq. (3.1), denoted as $\{x_n^{\texttt{ALG}}\}_{n\in[N]}$. We then show that $\texttt{ALG}(\mathcal{I})$ equals the objective value of the primal LP solution.

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

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

We then establish that the dual solution $\{u_n^{\texttt{ALG}}, \lambda_n^{\texttt{ALG}}\}_{n\in[N]}$ is feasible for the dual LP in Eq. (B.1).

In the final step, we establish that

$$D^{\texttt{ALG}} \leq \Big(1 + \ln\Big(\frac{v_{\max}}{v_{\min}}\Big)\Big) \cdot P^{\texttt{ALG}}.$$

After proving these steps, by weak duality we have

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

where $\texttt{OPT}(\mathcal{I})$ 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 2 follows.

First Step: Design of Primal Solution and $\texttt{ALG}(\mathcal{I}) = P^{\texttt{ALG}}$. We set the primal variables as $x_n^{\texttt{ALG}} = x_n$ for all $n \in [N]$. Let $\Delta^{(n)}_{P^{\texttt{ALG}}}$ denote the increase in the primal LP objective value resulting from updating $x_n^{\texttt{ALG}}$, and let $\Delta_n^{\texttt{ALG}(\mathcal{I})}$ denote the corresponding increase in the expected objective value of Algorithm 2 when a unit of resource is allocated to buyer n. By our update, we have

$$\Delta^{(n)}_{P^{\texttt{ALG}}} = v_n \cdot x_n^{\texttt{ALG}}.$$

Furthermore,

$$\begin{aligned} \Delta_n^{\texttt{ALG}(\mathcal{I})} &= v_n \cdot Pr[\text{a unit of resource is allocated to buyer } n] \\ &= v_n \cdot x_n. \end{aligned}$$

The final equality follows because the 1-(k, d)-OCA rounding scheme allocates a unit to buyer n with probability exactly x_n. Summing over buyers for all $n \in [N]$ establishes that $\texttt{ALG}(\mathcal{I}) = P^{\texttt{ALG}}$.

Second Step: Design of Dual Solution, $\{u_n^{\texttt{ALG}}, \lambda_n^{\texttt{ALG}}\}_{n\in[N]}$, and Feasibility of the Dual Solution. Let us initialize all dual variables to zero. For each rental request n, let x_n be the variable chosen by Algorithm 2. We then perform the following updates:

$$\lambda^{\texttt{ALG}}_{\nu_n^*} = \lambda^{\texttt{ALG}}_{\nu_n^*} + F \cdot \int_{\eta=\frac{y_n}{k}}^{\frac{y_n+x_n}{k}} \phi(\eta) d\eta, \tag{B.4}$$

$$u_n^{\texttt{ALG}} = F \cdot x_n \cdot \left(v_n - \phi\left(\frac{y_n + x_n}{k}\right)\right), \tag{B.5}$$

where $\nu_n^* = \max\{j \geq n | a_j < a_n + d\}$ 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. (B.2) corresponding to each buyer n is satisfied; that is,

$$u_n^{\texttt{ALG}} + \sum_{j=n}^{N} \lambda_j^{\texttt{ALG}} \cdot \mathbf{1}_{\{a_n+d>a_j\}} \geq v_n.$$

For the n-th buyer, the dual variable $u_n^{\texttt{ALG}}$ is updated according to Eq. (B.5), so that

$$u_n^{\texttt{ALG}} \geq F \cdot x_n \cdot \left(v_n - \phi\Big(\frac{y_n + x_n}{k}\Big)\right).$$

Next, we aim to prove that

$$\sum_{j=n}^{N} \lambda_j^{\texttt{ALG}} \cdot \mathbf{1}_{\{a_n + d > a_j\}} \geq F \cdot \int_0^{\frac{y_n + x_n}{k}} \phi(\eta) d\eta. \tag{B.6}$$

Assuming this inequality holds, we first show that the dual constraint in Eq. (B.2) is satisfied for each buyer n and then get back to the proof of the above inequality. Let us consider the following two cases.

Case 1: $\phi\Big(\frac{y_n+x_n}{k}\Big) < v_n$. It can be verified that following from Eq. (3.5), we have

$$x_n = \max\Big\{0, \min\Big\{1, k \cdot \phi^{-1}(v_n) - y_n\Big\}\Big\}.$$

Since we have $\phi\Big(\frac{y_n+x_n}{k}\Big) < v_n$, then it follows that $x_n = 1$. Therefore,

$$\begin{aligned}
u_n^{\texttt{ALG}} + \sum_{j=n}^{N} \lambda_j^{\texttt{ALG}} \cdot \mathbf{1}_{\{a_n + d > a_j\}} &\geq F \cdot \left(v_n - \phi\Big(\frac{y_n + x_n}{k}\Big)\right) + F \cdot \int_0^{\frac{y_n + x_n}{k}} \phi(\eta) d\eta \\
&\geq F \cdot \left(\int_{\frac{y_n+x_n}{k}}^{\phi^{-1}(v_n)} \phi(\eta) d\eta + \int_0^{\frac{y_n + x_n}{k}} \phi(\eta) d\eta\right) \\
&= F \cdot \int_0^{\phi^{-1}(v_n)} \phi(\eta) d\eta \\
&\geq v_n,
\end{aligned}$$

where the second inequality follows from the monotonicity of ϕ, and the last inequality follows from the design of ϕ in Theorem 3.2.

Case 2: $\phi\Big(\frac{y_n+x_n}{k}\Big) \geq v_n$. We can lower-bound the LHS of Eq. (B.2) as follows:

$$
\begin{aligned}
u_n^{\texttt{ALG}} + \sum_{j=n}^{N} \lambda_j^{\texttt{ALG}} \cdot \mathbf{1}_{\{a_n+d>a_j\}} &\geq F \cdot x_n \cdot \Big(v_n - \phi\Big(\frac{y_n+x_n}{k}\Big)\Big) + F \cdot \int_0^{\frac{y_n+x_n}{k}} \phi(\eta)d\eta \\
&\geq F \cdot \int_0^{\frac{y_n+x_n}{k}} \phi(\eta)d\eta \\
&\geq F \cdot \int_0^{\phi^{-1}(v_n)} \phi(\eta)d\eta \\
&\geq v_n,
\end{aligned}
$$

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

Thus, assuming Eq. (B.6) holds, the dual constraint in Eq. (B.2) is satisfied for each $n \in [N]$, proving the feasibility of the dual solution $\{u_n^{\texttt{ALG}}, \lambda_n^{\texttt{ALG}}\}$.

Let us now return to the proof of inequality Eq. (B.6). Define the set of buyers O_n as

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

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

$$B_n = \{1 \leq j \leq n \mid a_j + d > a_n, x_j > 0\}.$$

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

$$
\begin{aligned}
\sum_{j=n}^{N} \lambda_j^{\texttt{ALG}} \cdot \mathbf{1}_{\{a_n+d>a_j\}} &\geq \sum_{j \in O_n} \lambda_j^{\texttt{ALG}} \\
&= \sum_{j \in O_n} \sum_{m \in [n]} \Delta_m^{\lambda_j^{\texttt{ALG}}} \\
&\geq \sum_{m \in B_n} \Delta_m^{\lambda_{\nu_m^*}^{\texttt{ALG}}},
\end{aligned}
$$

where in above $\Delta_m^{\lambda_j^{\texttt{ALG}}}$ denotes the increase in the value of $\lambda_j^{\texttt{ALG}}$ due to the dual update in Eq. (B.5) performed for buyer m. In above, the first inequity follows from the definition of the set O_n and the second inequality follows from the fact that the buyer ν_m^* is in the set O_n for each buyer $m \in B_n$.

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

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

We have:

$$\begin{aligned}
\sum_{m \in B_n} \Delta_m^{\lambda_{\nu_m^*}^{\texttt{ALG}}} &\geq F \sum_{i=1}^{|B_n|} \int_{\eta=y_{n_i}}^{y_{n_i}+x_{n_i}} \phi(\eta) d\eta \\
&\geq F \sum_{i=1}^{|B_n|} \int_{\eta=\sum_{j=1}^{i-1} x_{n_j}}^{\sum_{j=1}^{i} x_{n_j}} \phi(\eta) d\eta \\
&= F \int_{\eta=0}^{\sum_{i=1}^{|B_n|} x_{n_i}} \phi(\eta) d\eta \\
&= F \int_{\eta=0}^{y_n+x_n} \phi(\eta) d\eta.
\end{aligned}$$

In the above, the first inequality follows from the update for $\lambda_{\nu_m^*}^{\texttt{ALG}}$ in Eq. (B.5) for each buyer m, and the second inequality holds because, by the definition of the set B_n, all buyers in B_n arriving prior to buyer n_i have rental requests overlapping buyer n_i; hence, the variable y_{n_i} is at least $\sum_{j=1}^{i-1} x_{n_j}$ at the arrival of buyer n_i.

Putting together the results obtained in the above two cases, the inequality in Eq. (B.6) follows.

Third step: $D^{\texttt{ALG}} \leq F \cdot P^{\texttt{ALG}}$**.** Instead of proving the overall inequality $D^{\texttt{ALG}} \leq F \cdot P^{\texttt{ALG}}$ directly, we show that for each buyer n

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

where $\Delta_n^{D^{\texttt{ALG}}}$ denotes the increase in the dual objective value after updating the dual variables for buyer n via Eqs. (B.4) and (B.5), and $\Delta_n^{P^{\texttt{ALG}}}$ denotes the increase in the primal objective value after updating the primal LP solution for buyer n by setting $x_n^{\texttt{ALG}} = x_n$.

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

Case 1: $x_n = 1$ In this case, we have

$$\begin{aligned}
\Delta_n^{D^{\texttt{ALG}}} &= \Delta_n^{u_n^{\texttt{ALG}}} + k\Delta_n^{\lambda_{\nu_n^*}^{\texttt{ALG}}} \\
&= F \cdot x_n \cdot \left(v_n - \phi\left(\frac{y_n + x_n}{k}\right)\right) + kF \cdot \int_{\eta=\frac{y_n}{k}}^{\frac{y_n+x_n}{k}} \phi(\eta)d\eta \\
&= F \cdot \left(v_n - \phi\left(\frac{y_n + 1}{k}\right)\right) + kF \cdot \int_{\eta=\frac{y_n}{k}}^{\frac{y_n+1}{k}} \phi(\eta)d\eta \\
&\leq F \cdot \left(v_n - \phi\left(\frac{y_n + 1}{k}\right)\right) + kF \cdot \frac{1}{k}\phi\left(\frac{y_n + 1}{k}\right) \\
&= F \cdot v_n = F\Delta_n^{P^{\texttt{ALG}}},
\end{aligned}$$

where in the above the first equality follows from the objective value of the dual LP in Eq. (B.1). The second equality follows from the dual updates done in Eq. (B.4) and Eq. (B.5). Then, the first inequality follows from the fact that ϕ is an increasing function.

Case 2: $x_n < 1$ In this case, if $x_n = 0$, then we have $\Delta_n^{D^{\texttt{ALG}}} = \Delta_n^{P^{\texttt{ALG}}} = 0$. Otherwise, if $x_n \neq 0$, then by Eq. (3.5) we have

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

Since $x_n < 1$, we must have

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

It follows that

$$\begin{aligned}
\Delta_n^{D^{\texttt{ALG}}} &= \Delta_n^{u_n^{\texttt{ALG}}} + k\Delta_n^{\lambda_{\nu_n^*}^{\texttt{ALG}}} \\
&= F \cdot x_n \cdot \left(v_n - \phi\left(\frac{y_n + x_n}{k}\right)\right) + kF \cdot \int_{\eta=\frac{y_n}{k}}^{\frac{y_n+x_n}{k}} \phi(\eta)d\eta \\
&= kF \cdot \int_{\eta=\frac{y_n}{k}}^{\frac{y_n+x_n}{k}} \phi(\eta)d\eta, \\
&\leq Fk \cdot \frac{x_n}{k} \cdot \phi\left(\frac{y_n + x_n}{k}\right) \\
&= F \cdot v_n \cdot x_n \\
&= F\Delta_n^{P^{\texttt{ALG}}}.
\end{aligned}$$

In the above, the first inequality follows from the fact that the function ϕ is increasing.

Thus, combining the results from both cases, we obtain

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

which completes the proof of the third step and thus the proof for Theorem 3.2.

B.2 Proof of Proposition 3.2

Following the same proof structure as in [SJTB24] for the online selection problem, we can prove the lower bound $1 + \ln(\frac{v_{\max}}{v_{\min}})$ on the competitiveness of every online algorithm for `kRental-fixed`.

We can design a set of hard instances for the `kRental-fixed` problem similar to [SJTB24]. Let $\mathcal{A}(k, v)$ denote a batch of k identical buyers, 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 buyer 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 buyers—from those in the first batch $\mathcal{A}(k, v_1)$ to the last buyer 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 buyer in batch $\mathcal{A}(k, v_i)$, it cannot be reallocated to any buyer arriving afterwards, because the unit will only become available after d time steps, while subsequent buyers 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 [SJTB24], 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 $1 + \ln(\frac{v_{\max}}{v_{\min}})$ 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 $1 + \ln(\frac{v_{\max}}{v_{\min}})$.

B.3 Proof of Proposition 3.1

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

$$R_n^{(i)} = \Big\{ \big(p_j, \max\{1, p_j + x_j\}, j\big) \big| j \in B_n^{(i)} \Big\} \cup \Big\{ \big(0, p_j + x_j - 1, j\big) \big| j \in B_n^{(i-1)}, p_j + x_j > 1 \Big\}, \quad \text{(B.7)}$$

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

$$B_n^{(i)} = \Big\{ j \in [n] \mid a_j + d > a_n, m_j = i, x_j \neq 0 \Big\}.$$

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 $\Big(p_j, \max\{1, p_j + x_j\}, j\Big)$ 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 $(p_j, \max\{1, p_j + x_j\}]$, Procedure 3 allocates ball $m_j = i$ to player j.

Similarly, if $\Big(0, p_j + x_j - 1, j\Big)$ belongs to $R_n^{(i)}$ and $j \in B_n^{(i-1)}$, then upon the arrival of player j, if $r \in [0, p_j + x_j - 1)$, Algorithm 3 allocates ball $m_j + 1 = i$ to player j. Thus, the element $\Big(0, p_j + x_j - 1, j\Big)$ is added to $R_n^{(i)}$ to reflect this allocation decision.

In what follows, we show that if Proposition 3.1 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 3.1:

Case 1: $p_n + x_n < 1$**.** If Proposition 3.1 holds, then whenever $r \in [p_n, p_n + x_n)$, ball m_n remains available at the arrival of player n. Suppose the lemma fails in this case. Then, for some $r \in [p_n, p_n + x_n)$, ball m_n is unavailable. Hence, there must exist a player $j < n$ such that if $r \in [s_1, e_1)$, with $[s_1, e_1) \cap [p_n, p_n + x_n) \neq \varnothing$, ball m_n was allocated to j. Thereby, in this case Proposition 3.1 doesn't hold, by the definition of $R_n^{(m_n)}$, the set $R_n^{(m_n)}$ contains two triplets (s_1, e_1, j) and $(p_n, p_n + x_n, n)$ such that range $[s_1, e_1)$ and $[p_n, p_n + x_n)$ overlap.

Case 2: $p_n + x_n \geq 1$**.** A similar argument shows that if Proposition 3.1 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 intervals $[s_1, e_1)$ and $[s_2, e_2)$ intersect.

We next 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 B.1 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 B.1. *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* *three forms arises:*

(i) *If $i = m_n$ and $x_n + p_n < 1$, then for some $j \in \mathbb{N}$, $R_n^{(i)}$ takes the form*

$$R_n^{(i)} = \Big\{ \big(s_1, s_2, t_1\big), \ldots, (s_j = p_n, s_{j+1} = p_n + x_n, t_j), (s_{j+1} + 1 - l_n^{(i)}, s_{j+2}, t_{j+1}), \ldots, (s_q, s_{q+1} = 1, t_q) \Big\},$$

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 $x_n + p_n \geq 1$, $R_n^{(i)}$ takes the form*

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

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 $x_n + p_n > 1$ then $R_n^{(i)}$ takes the form*

$$R_n^{(i)} = \Big\{ (s_1 = 0, s_2 = p_n + x_n - 1, t_1 = n), \big(s_2, s_3, t_2\big), (s_3, s_4, t_3), \ldots, (s_q, s_{q+1} = 1, t_q) \Big\},$$

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 $x_n + p_n \leq 1$ or $i \in [k] - \{m_n, m_n + 1\}$*

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

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 $x_n + p_n < 1$ then $R_1^{(1)}$ takes the first form $j = 1$, otherwise it is in the second form and all other sets $R_1^{(i)}$ are empty. Therefore, the proposition holds for $N = 1$.

Induction Hypothesis: 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 B.1.

Inductive Step: 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 discussion below, we examine the two cases that can arise depending on whether an element is added to or removed from the set $R_M^{(i)}$.

Case 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

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

for some values $\{s_i\}$ and $\{t_i\}$ satisfying the conditions of the first form in Proposition B.1.

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 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 B.1.

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.

Case 2 (element is added)

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

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

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

Suppose a new element $(s, e, M+1)$ is added at the arrival time of the $(M+1)$-th player. By the design of Procedure 3, 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 $R^{(i)}_{M+1}$ remains in the proper form (i.e., either in the first or second form), we must have

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

We now formalize and prove this requirement.

Lemma B.1. *Consider the case where the set $R^{(i)}_M$ 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*

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

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

Proof. Consider the stage when the algorithm has finished processing the t_q-th player and has added the element $(s_q, s_{q+1} = 1, t_q)$ to $R^{(i)}_{t_q}$. By design, at this stage the pointer p_{t_q+1} reaches 1, and the pointer m_{t_q} is incremented. Between the processing of the t_q-th and t_1-th players, the pointer m_n is incremented $k-1$ times so that it again points to ball i, and a new element (s_1, s_2, t_1) is added to $R^{(i)}_{t_1}$. For the pointer m_n to be incremented $k-1$ times, there exists a sequence of players from the t_q-th to the t_1-th player such that

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

Next, it follows that if

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

then

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

which contradicts the constraint in Eq. (3.6). Therefore, we must have

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

Thus, we complete the proof of Lemma B.1. $\square$

Thus, 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 at arrival time of player $M+1$, a new element is added to $R_{M+1}^{(i)}$, then it must be the case that $k = 1$, and the newly added element is $(p_{M+1},\ p_{M+1} + x_{M+1},\ M+1)$. Following the same analysis, if the range $(p_{M+1},\ p_{M+1} + x_{M+1}]$ of this newly added element overlaps with any other element in $R_{M+1}^{(i)}$, we obtain a contradiction with the constraint in Eq. (3.6).

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

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

Suppose a new element (s, e, t) is added at the arrival time of player $M+1$. Then, by the design of Procedure 3,

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

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

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. (3.6).

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

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

Suppose a new element (s, e, t) is added at time $M + 1$. By the design of Algorithm 3, 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. (3.6). □

Thus, by completing the proof of Proposition B.1, we have shown 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 $[s_2, e_2)$. Hence, the proof of Proposition 3.1 follows.

B.4 Proof of Theorem 3.1

We consider two cases based on the relation between $1 - p_n$ and x_n.

Case I: $1 - p_n \geq x_n$. In this case, the algorithm assigns ball m_n to player n if ball m_n is available and $r \in [p_n, p_n + x_n)$. 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 $r \in [p_n, p_n + x_n)$. 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 B.1 ensures that the range $[p_n, p_n + x_n)$ does not overlap with any other range in $R_n^{(m_n)}$. Thus, if the random sample $r \in [p_n, p_n + x_n)$, 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,

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

Hence, when $x_n < 1 - p_n$, player n gets ball m_n with probability x_n.

Case II: $1 - p_n < x_n$. 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 $r \in [0, x_n - 1 + p_n)$. 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 $r \in [0, x_n - 1 + p_n)$. Then the total probability that a ball is allocated to player n is given by

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

Based on Proposition B.1 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

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

Since the ranges $[p_n, 1]$ and $[0, x_n - 1 + p_n)$ do not overlap (because $x_n \leq 1$), we have

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

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

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

Thus, in both cases, each player n gets a ball with the desired probability x_n, which proves that Algorithm 3 is a lossless online scheme.

Appendix C

Proofs of Chapter 4

C.1 Proof of Proposition 4.1

We apply the LP-free certificate approach developed by [GIU0] to upper-bound the competitive ratio of Algorithm 4 for any given instance $I \in \mathcal{I}^{\texttt{kRental-variable}}$ of `kRental-variable` problem. Using this approach, we define a system of linear constraints such that the existence of a feasible solution certifies α-competitiveness of the algorithm.

Let $\{\{u_n, \lambda_n^{(i)}\}_{n\in[N]}, \theta_i\}_{i\in[k]}$ be the set of variables in our system. The constraints are given by:

$$\sum_{n\in[N]} u_n + \sum_{i\in[k]} \theta_i \leq \alpha \cdot \texttt{ALG}(I), \tag{C.1}$$

$$\theta_i + \sum_{n\in P_i} u_n \geq \sum_{n\in P_i} d_n = \texttt{OPT}_i, \qquad \forall i \in [k], \tag{C.2}$$

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

$$\begin{aligned}
\texttt{OPT} &= \sum_{i=1}^{k} \texttt{OPT}_i \\
&\leq \sum_{i=1}^{k} \Big(\theta_i + \sum_{n\in P_i} u_n\Big) \\
&\leq \sum_{i\in[k]} \theta_i + \sum_{n\in[N]} u_n \\
&\leq \alpha \cdot \texttt{ALG}(I),
\end{aligned}$$

where in the derivation above, the first inequality follows from Eq. (C.1), the second inequality holds because the sets $\{P_i\}$ are pairwise disjoint, and the final inequality follows from Eq. (C.6). 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 buyer in instance I, we update the variables as follows. For each rental request n, let x_n be the fractional allocation variable determined by Algorithm 4. We then perform the following updates:

$$u_n = \frac{\alpha}{3} d_n x_n, \tag{C.3}$$

$$\lambda_j^{(i_n^*)} = \lambda_j^{(i_n^*)} + \frac{\alpha}{3}\big(a_{j+1} - a_j\big)x_n, \quad \forall j, n \leq j < \nu_n^*, a_j < a_n + d_n, \tag{C.4}$$

$$\lambda_{\nu_n^*}^{(i_n^*)} = \lambda_{\nu_n^*}^{(i_n^*)} + \frac{\alpha}{3}\Big(2d_n - \big(a_{\nu_n^*} - a_n\big)\Big)x_n, \tag{C.5}$$

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

Remark C.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 buyer n, and the set of variables $\lambda_j^{(i)}$ represents the price of unit i at the arrival time of buyer j. Depending on the fractional allocation x_n, which is allocated from unit i_n^* to buyer n, the utility u_n increases accordingly. Furthermore, we raise the price of unit i_n^* at the arrival time of every buyer $j < \nu_n^*$ whose request interval overlaps with that of buyer n. This price adjustment is intended for any buyer arriving after buyer n whose rental duration finishes before that of buyer n and who might be rejected because a fraction of unit i_n^* has already been allocated to buyer n. In addition, we also increase the price of unit i_n^* for buyer ν_n^*, which is the last arriving buyer in the horizon whose request interval overlaps with that of buyer n. This adjustment accounts for buyers who might arrive very close to the end of buyer n's request interval and whose requests may be rejected because the algorithm has already allocated a fraction of unit i_n^* to buyer n.*

First Constraint of the System. Based on the construction of the variables detailed above, we now verify the constraint Eq. (C.1).

Let $\Delta_n^{\texttt{ALG}}$ denote the increase in the expected objective value of Algorithm 4 after processing buyer n. We show that $\Delta_n^{\texttt{ALG}} = d_n \cdot x_n$. Algorithm 4 defines $y_n^{(i)}$ as in Eq. (4.4), which sums the probabilities that unit i has been allocated to previous buyers. Consequently, at the arrival of buyer

n, resource i is available with probability $1 - y_n^{(i)}$. Therefore,

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

Summing over all buyers $n \in [N]$, $\mathtt{ALG}(I) = \sum_{n\in[N]} \Delta_n^{\mathtt{ALG}} = \sum_{n\in[N]} d_n x_n$.

Next, let Δ_n^{RHS} denote the increase in the right-hand side of Eq. (C.1) after updating the variables of the system according to Eqs. (C.3)–(C.5) for the n-th rental request. It follows directly that

$$\Delta_n^{\text{RHS}} = \alpha d_n x_n.$$

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

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

Combining this with our previous result on the expected increase in the objective of the algorithm, we conclude that the first constraint, Eq. (C.1), is satisfied.

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

$$\theta_i + \sum_{n\in P_i} u_n \geq \alpha \cdot \sum_{n\in P_i} d_n = \alpha \cdot \mathtt{OPT}_i, \tag{C.6}$$

where the value of θ_i can be lower bounded by:

$$\theta_i = \sum_{n\in[N]} \lambda_n^{(i)} \geq \sum_{n\in P_i} \sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n + d_n > a_j\}}.$$

The inequality holds because, for each buyer $n \in P_i$, the set $\{j \in [N] \mid a_n + d_n > a_j\}$ cannot intersect with any other buyer's set. Otherwise, the optimal offline algorithm would have allocated unit i to two buyers at the same time, which is infeasible. Hence, for the left-hand side of Eq. (C.6), we have

$$\theta_i + \sum_{n\in P_i} u_n \geq \sum_{n\in P_i} \sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}} + \sum_{n\in P_i} u_n \geq \sum_{n\in P_i} \Big(\sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}} + u_n \Big).$$

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

$$\sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}} + u_n \geq \alpha \cdot d_n.$$

In what follows, we first derive a lower bound on the left-hand side of the above inequality by separately bounding $\sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}}$ and u_n. Then, invoking the constraints placed on the price function ϕ in Proposition 4.1, we conclude that this bound is at least $\alpha \cdot d_n$, thus ensuring that Eq. (C.6) is satisfied.

Let us define

$$\Lambda_n^{(i)} = \sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}}.$$

Lower-bounding Term $\Lambda_n^{(i)}$. Let us define the set of buyers $B_n^{(i)}$ by

$$B_n^{(i)} = \big\{1 \leq j < n \mid a_j + d_j > a_n, x_j > 0, i_j^* = i\big\}. \tag{C.7}$$

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

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

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

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

$$\begin{aligned} c_1 &= \arg\min_{j \in C_n^{(i)}} \{a_j + d_j\}, \\ c_l &= \arg\min_{\substack{j \in C_n^{(i)} \\ a_j > a_{c_{l-1}}}} \{a_j + d_j\}, \qquad 1 < l \leq L. \end{aligned}$$

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

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

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

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

Let $\Delta^{(j)}_{\Lambda_n^{(i)}}$ denote the increase in $\Lambda_n^{(i)}$ after processing the rental request of buyer j and updating the variables according to Eqs. (C.3)–(C.5). From the way we perform the updates, it follows that

$$\begin{aligned}
\sum_{j\in C_n^{(i)}} \Delta^{(j)}_{\Lambda_n^{(i)}} &\geq \sum_{l=1}^{L} \sum_{j\in \mathcal{C}_l} \frac{\alpha}{3} x_j \Big(d_j + \big(a_j + d_j - a_n\big)\Big) \\
&\geq \sum_{l=1}^{L} \sum_{j\in \mathcal{C}_l} \frac{\alpha}{3} x_j \Big(d_{c_l} + \big(a_{c_l} + d_{c_l} - a_n\big)\Big) \\
&\geq \sum_{l=1}^{L} \frac{\alpha}{3} \big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1}\Big),
\end{aligned}$$

where the first inequality follows from the dual updates in Eqs. (C.4)–(C.5), the second inequality relies on the definition of c_l and $\mathcal{C}_l$ (under which $d_j \leq d_{c_l}$ and $a_j + d_j \geq a_{c_l} + d_{c_l}$ for each $j \in \mathcal{C}_l$), 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:

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

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

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

Moreover,

$$\begin{aligned}
&\sum_{l=1}^{L} \frac{\alpha}{3} \big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big) \\
\geq\ & \frac{\alpha}{3} \cdot z'_1 \phi(z'_1) + \sum_{l=2}^{L} \big(z'_l - z'_{l-1}\big)\big(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\big).
\end{aligned} \tag{C.8}$$

Proof. **Part-I (Proof of $d_{c_l} \geq \phi(z'_l)$).** We show $d_{c_l} \geq \phi\big(z'_l\big)$ 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 buyer c_1 is at least z_1. Since $x_{c_1} \neq 0$, it follows that

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

Assume the statement holds for any number of buyers 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

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

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

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

Otherwise, suppose

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

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

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

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

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

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 $\phi\big(z'_l\big) \leq d_{c_l}$ for all $l \in \{1, \ldots, M-1\}$.

Part-II (Proof of (C.8)). We again use induction on L to prove

$$\begin{aligned}&\sum_{l=1}^{L} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big)\\ \geq\ &\frac{\alpha}{3} z'_1 \phi(z'_1) + \sum_{l=2}^{L}\big(z'_l - z'_{l-1}\big)\Big(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\Big).\end{aligned}$$

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 $\{z_1, \dots, z_{L-2}\} \cup \{z'_{L-1}\}$, we have

$$\begin{aligned}
&\sum_{l=1}^{L-2} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1}\Big) + \frac{\alpha}{3}\big(z'_{L-1} - z_{L-2}\big)\Big(2d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\Big) \\
\geq& \frac{\alpha}{3} z'_1 d_{c_1} + \sum_{l=2}^{L-1} \big(z'_l - z'_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big).
\end{aligned}$$

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

$$\begin{aligned}
&\sum_{l=1}^{L-2} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1}\Big) + \frac{\alpha}{3}\big(z'_{L-1} - z_{L-2}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big) + \\
&\qquad \frac{\alpha}{3}\big(z_L - z'_{L-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big) \\
\geq& \frac{\alpha}{3} \cdot z'_1 \cdot d_{c_1} + \sum_{l=2}^{L} (z'_l - z'_{l-1}) \cdot (2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}).
\end{aligned}$$

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

$$\begin{aligned}
&\sum_{l=1}^{L-2} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big) + \frac{\alpha}{3}\big(z'_{L-1} - z_{L-2}\big)\Big(2d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\Big) + \\
&\qquad \frac{\alpha}{3}\big(z_L - z'_{L-1}\big)\Big(2d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\Big)) \\
\leq& \sum_{l=1}^{L-2} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big) + \frac{\alpha}{3}\big(z_{L-1} - z_{L-2}\big)\Big(2d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\Big) + \\
&\qquad \frac{\alpha}{3}\big(z_L - z_{L-1}\big)\Big(2d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\Big)) \\
=& \sum_{l=1}^{L} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\Big(2d_{c_l} + a_{c_l} - a_{c_1} - d_{c_1}\Big),
\end{aligned}$$

where the inequality follows since $\Big(2d_{c_{L-1}} + a_{c_{L-1}} - a_{c_1} - d_{c_1}\Big) \leq \Big(2d_{c_L} + a_{c_L} - a_{c_1} - d_{c_1}\Big)$. Thus in this case the inequality in Eq. (C.8) 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 (C.8). $\square$

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

$$\begin{aligned}
\sum_{j\in C_n^{(i)}} \Delta^{(j)}_{\Lambda_n^{(i)}} &\geq \sum_{l=1}^{L} \frac{\alpha}{3}\big(z_l - z_{l-1}\big)\big(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\big) \\
&\geq \frac{\alpha}{3}\Big[z_1'\phi(z_1') + \sum_{l=2}^{L}\big(z_l' - z_{l-1}'\big)\big(d_{c_l} + a_{c_l} + d_{c_l} - a_{c_1} - d_{c_1}\big)\Big] \\
&\geq \frac{\alpha}{3}\Big[z_1'\phi(z_1') + \sum_{l=2}^{L}\big(z_l' - z_{l-1}'\big)\big(2\phi(z_l') - \phi(z_1')\big)\Big] \\
&\geq \frac{\alpha}{3}\Big[z_1'\phi(z_1') + \int_{\eta=z_1'}^{z_L'}\big(2\phi(\eta) - \phi(z_1')\big)d\eta\Big] \\
&= \frac{2\alpha}{3}z_1'\phi(z_1') - \frac{\alpha}{3}z_L'\phi(z_1') + \int_{z_1'}^{z_L'}\frac{2\alpha}{3}\phi(\eta)d\eta,
\end{aligned}$$

where the second and third inequalities follow from Lemma C.1, and the fourth inequality holds because ϕ is an increasing function.

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

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

The above inequality holds because, for each buyer 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. (C.4)–(C.5), for each such j, $\Delta^{(j)}_{\Lambda_n^{(i)}}$ is at least $\frac{\alpha}{3}d_n x_j$. Furthermore,

$$\sum_{j\in B_n^{(i)}\setminus C_n^{(i)}} x_j \geq \max\Big\{0, y_n^{(i_n^*)} - z_L'\Big\}.$$

Lower-Bounding the Term u_n. From the update rule in Eq. (C.3), for each buyer n we have:

$$\begin{aligned}
u_n &\geq \frac{\alpha}{3}d_n x_n \\
&\geq \frac{\alpha}{3}d_n \max\Big\{0, \phi^*\big(d_n\big) - y_n^{(i_n^*)}\Big\},
\end{aligned}$$

where the second inequality follows from the way Algorithm 4 sets the value of x_n in Eq. (4.5).

Combining bounds obtained for the LHS of Eq. (C.6), we obtain

$$\begin{aligned}
u_n + \sum_{j=n}^{N} \lambda_j^{(i)} \mathbf{1}_{\{a_n+d_n>a_j\}} &\geq \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - y_n^{(i_n^*)}\Big\} + \sum_{j\in B_n^{(i)}} \Delta^{(j)}_{\Lambda_n^{(i)}} \\
&= \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - y_n^{(i_n^*)}\Big\} + \sum_{j\in C_n^{(i)}} \Delta^{(j)}_{\Lambda_n^{(i)}} + \sum_{j\in B_n^{(i)}\setminus C_n^{(i)}} \Delta^{(j)}_{\Lambda_n^{(i)}} \\
&\geq \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - y_n^{(i_n^*)}\Big\} + \frac{2\alpha}{3} z_1' \phi\big(z_1'\big) - \frac{\alpha}{3} z_L' \phi\big(z_1'\big) \\
&\qquad + \int_{z_1'}^{z_L'} \frac{2\alpha}{3}\phi(\eta) d\eta + \frac{\alpha}{3} d_n \max\Big\{0, y_n^{(i_n^*)} - z_L'\Big\} \\
&= \frac{2\alpha}{3} z_1' \phi\big(z_1'\big) - \frac{\alpha}{3} z_L' \phi\big(z_1'\big) + \int_{z_1'}^{z_L'} \frac{2\alpha}{3}\phi(\eta) d\eta \\
&\qquad + \frac{\alpha}{3} d_n \max\Big\{0, y_n^{(i_n^*)} - z_L'\Big\} + \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - y_n^{(i_n^*)}\Big\}. \\
&\geq \frac{2\alpha}{3} z_1' \phi\big(z_1'\big) - \frac{\alpha}{3} z_L' \phi\big(z_1'\big) + \int_{z_1'}^{z_L'} \frac{2\alpha}{3}\phi(\eta) d\eta \\
&\qquad + \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - z_L'\Big\} \\
&\geq d_n,
\end{aligned}$$

where the last inequality follows from the lemma given below. Consequently, we conclude that the second set of constraints in Eq. (C.6) is satisfied by the design of system variables in Eqs. (C.3)–(C.5). Therefore, if the increasing pricing function ϕ satisfies the system of constraints in Proposition 4.1, it establishes the α-competitiveness of Algorithm 4.

Lemma C.2. *If the ϕ function satisfy the constraints in Eqs. (4.6)–(4.7) then*

$$\frac{2\alpha}{3} z_1' \phi\big(z_1'\big) - \frac{\alpha}{3} z_L' \phi\big(z_1'\big) + \int_{z_1'}^{z_L'} \frac{2\alpha}{3}\phi(\eta) d\eta + \frac{\alpha}{3} d_n \max\Big\{0, \phi^*\big(d_n\big) - z_L'\Big\} \geq d_n,$$
$$\forall d_n \in [d_{\min}, d_{\max}], z_1' \in \big(0, z_L'\big], z_L' \in \big(0, \phi^*(d_{min})\big].$$

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

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

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

$$\frac{\partial \mathfrak{L}}{\partial z'_1} = \frac{\alpha}{3}\phi'(z'_1)\Big(2z'_1 - z'_L\Big).$$

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

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

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

$$\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}z'_L\Big(d_n - \phi(z'_L)\Big)$$

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:

$$\begin{aligned}
\mathfrak{L}(0, z'_L, d_n) &= \int_0^{z'_L} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\Big\{0, \phi^*\big(d_n\big) - z'_L\Big\} - \frac{\alpha}{3}z'_L\phi\big(z'_1\big) \\
&\geq \int_{z'_L/2}^{z'_L} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n \max\Big\{0, \phi^*\big(d_n\big) - z'_L\Big\} = L(z'_L/2, z'_L, d_n),
\end{aligned}$$

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

$$\begin{aligned}
\mathfrak{L}(z_1', z_L', d_n) &\geq \min\Big\{\mathfrak{L}\Big(\frac{z_L'}{2}, z_L', d_n\Big), \quad \mathfrak{L}(z_L', z_L', d_n)\Big\} \\
&= \min\Bigg\{\frac{\alpha}{3} d_n \phi^*(d_n) - \frac{\alpha}{3} z_L'\Big(d_n - \phi(z_L')\Big), \\
&\qquad \int_{z_L'/2}^{z_L'} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3} d_n \max\{0, \phi^*(d_n) - z_L'\}\Bigg\} \\
&\geq d_n,
\end{aligned}$$

where the last inequality follows from the constraints in Eq. (4.6) and Eq. (4.7). □

C.2 Proof of Proposition 4.2

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

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

the following constraints for ϕ are satisfied:

$$\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3} d_n\Big(\phi^*(d_n) - 2y_1\Big) \geq d_n, \tag{C.9}$$

$$\frac{\alpha}{3} d_n \phi^*(d_n) - \frac{\alpha}{3} y_2\Big(d_n - \phi(y_2)\Big) \geq d_n. \tag{C.10}$$

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

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

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

we have $\phi^*(d_n) = \epsilon \cdot m$ for some $m \in \Big\{1, 2, \ldots, \lceil \frac{1}{\epsilon} \rceil\Big\}$ because of the constraints in Eq. (4.8c) and Eq. (4.8d).

Thus, for any $y_1 \leq \frac{\phi^*(d_n)}{2} = \frac{\epsilon \cdot m}{2}$, the left-hand side of the constraint in Eq. (C.9) can be lower-bounded as follows:

$$
\begin{aligned}
&\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\Big(\phi^*(d_n) - 2y_1\Big)\\
=&\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\Big(\epsilon\cdot m - 2y_1\Big)\\
\geq&\sum_{j=\lfloor\frac{y_1}{\epsilon}\rfloor\cdot\epsilon}^{2\lfloor\frac{y_1}{\epsilon}\rfloor\cdot\epsilon} \frac{2\alpha}{3}\Big(\phi\big(\epsilon\cdot j\big) - \phi\big(\epsilon\cdot(j-1)\big)\Big)\epsilon + \frac{\alpha}{3}\phi\big(\epsilon\cdot(m-1)\big)\Big(\epsilon\cdot m - 2\epsilon\cdot\Big(\lfloor\frac{y_1}{\epsilon}\rfloor+1\Big)\Big)\\
=&\sum_{j=i\cdot\epsilon}^{2i\cdot\epsilon} \frac{2\alpha}{3}\Big(\phi\big(\epsilon\cdot j\big) - \phi\big(\epsilon\cdot(j-1)\big)\Big)\epsilon + \frac{\alpha}{3}\phi\big(\epsilon\cdot(m-1)\big)\Big(\epsilon\cdot m - 2\epsilon\cdot(i+1)\Big)\\
\geq&\phi\big(\epsilon\cdot m\big) \geq d_n,
\end{aligned}
$$

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. (4.8e) (since $i \leq \frac{m}{2}$), and the last inequality follows from the definition of ϕ^* and m.

Thus, the first constraint in Eq. (C.9) is satisfied for every $d_n \in [d_{\min}, d_{\max}]$ and every $y_1 \in \Big[0, \frac{\phi^*(d_n)}{2}\Big]$.

Next, let us prove that the constraint in Eq. (4.7). The LHS of that constraint can be lower-bounded as follows:

$$
\begin{aligned}
&\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}y_2\Big(d_n - \phi(y_2)\Big)\\
&\geq \frac{\alpha}{3}\phi(\epsilon\cdot(m-1))\epsilon\cdot m - \frac{\alpha}{3}\epsilon\cdot\Big(\lfloor\frac{y_2}{\epsilon}\rfloor+1\Big)\Big(\phi(\epsilon\cdot m) - \phi(\lfloor\frac{y_2}{\epsilon}\rfloor)\Big)\\
&= \frac{\alpha}{3}\phi(\epsilon\cdot(m-1))\epsilon\cdot m - \frac{\alpha}{3}\epsilon\cdot(i+1)\Big(\phi(\epsilon\cdot m) - \phi((i+1)\cdot\epsilon)\Big)\\
&\geq \phi(\epsilon\cdot m) \geq d_n,
\end{aligned}
$$

where $i = \lfloor\frac{y_2}{\epsilon}\rfloor$, the first inequality follows from simple algebra, the second inequality follows from the constraint in Eq. (4.8f) (since $y_2 \in [0, \phi^*(d_n)]$, thus $i \leq m$), and the final inequality follows from the definition of the ϕ function. Thus, the constraint in Eq. (C.10) is satisfied for every $d_n \in [d_{\min}, d_{\max}]$ and for every $y_2 \in \Big[0, \phi^*(d_n)\Big]$.

C.3 Proof of Corollary 4.1

We now prove that for the given design of the ϕ function, the constraints in Proposition 4.1 (see Eqs. (4.6) and (4.7)) are satisfied when

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

Proof for Eq. (4.6). The left-hand side of Eq. (4.6) can be lower-bounded as follows:

$$\begin{aligned}
&\int_{\eta=y_1}^{2y_1} \frac{2\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\Big(\phi^*(d_n) - 2y_1\Big) \\
\geq &\int_{\eta=0}^{2y_1} \frac{\alpha}{3}\phi(\eta)d\eta + \frac{\alpha}{3}d_n\Big(\phi^*(d_n) - 2y_1\Big) \\
\geq &\int_{\eta=0}^{\phi^*(d_n)} \frac{\alpha}{3}\phi(\eta)d\eta \\
\geq & d_n,
\end{aligned}$$

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.

Proof for Eq. (4.7). The left-hand side of Eq. (4.7) can be lower-bounded as:

$$\begin{aligned}
&\frac{\alpha}{3}d_n\phi^*(d_n) - \frac{\alpha}{3}y_2\Big(d_n - \phi(y_2)\Big) \\
= &\frac{\alpha}{3}y_2\phi(y_2) + \frac{\alpha}{3}d_n\Big(\phi^*(d_n) - y_2\Big) \\
\geq &\int_{\eta=0}^{\phi^*(d_n)} \frac{\alpha}{3}\phi(\eta)d\eta \\
\geq & d_n,
\end{aligned}$$

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 obtain competitiveness $3 \cdot \Big(1 + \ln \frac{d_{\max}}{d_{\min}}\Big)$ for the give design in Corollary.

C.4 Proof of Proposition 4.3

Following the same proof structure as in [SJTB24] 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. Thus, based on Corollary 4.1, we can claim that Algorithm 4 is order-optimal.

We can design a set of hard instances for the `kRental-variable` problem similar to [SJTB24]. Let $\mathcal{A}(k, d)$ denote a batch of k identical buyers, 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 buyer 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 buyers—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 in the proof of Lemma 2.3 in [SJTB24], 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.

Appendix D

Proofs of Chapter 5

D.1 Proof of Theorem 5.1

Given an instance $I = \{v_{n,j}\}_{n\in[N],j\in[J]}$ of the `Matching-BW` problem and a realization of j posted prices $\boldsymbol{p} = \{p_j\}_{j\in[J]}$, let $\{X_{n,j}\}_{n\in[N],j\in[J]}$ denote the online solution of rStatic-matching. Then the utility of buyer n and the total revenue earned by selling resource j can be determined by

$$\begin{aligned} u_n &= \sum\nolimits_{j\in[J]} X_{n,j}[v_{n,j} - p_j], \\ r_j &= \sum\nolimits_{n\in[N]} X_{n,j} p_j. \end{aligned}$$

The expected social welfare of rStatic-matching can be lower bounded by the total expected revenue and the total expected utility of the buyers that make purchases in the offline optimal solution, i.e.,

$$\begin{aligned} \mathbb{E}[\texttt{ALG}(\boldsymbol{p}, I)] &= \sum\nolimits_{n\in[N]} \mathbb{E}[u_n] + \sum\nolimits_{j\in[J]} \mathbb{E}[r_j] \\ &\geq \sum\nolimits_{j\in[J]} \left[\sum\nolimits_{n\in\mathcal{N}_j^*} \mathbb{E}[u_n] + \mathbb{E}[r_j]\right], \end{aligned}$$

where $\mathcal{N}_j^* \subseteq [N]$ is the set of buyers who decide to purchase resource j in the offline optimal solution. The offline social welfare can be denoted by $\texttt{OPT}(I) = \sum_{j\in[J]} \sum_{n\in\mathcal{N}_j^*} v_{n,j}$. Thus, to prove Theorem 5.1, it is sufficient to show that

$$\sum\nolimits_{n\in\mathcal{N}_j^*} \mathbb{E}[u_n] + \mathbb{E}[r_j] \geq \frac{1}{\alpha_j} \sum\nolimits_{n\in\mathcal{N}_j^*} v_{n,j}, \forall j \in [J]. \tag{D.1}$$

For each instance I, let $I^{(n,j)}$ denote a modified instance, in which the valuation $v_{n,j}$ of buyer

n for resource j is set to 0, forcing buyer n not to purchase resource j. Given j posted prices $\boldsymbol{p}$, let $u_n^{(n,j)}$ denote the utility of buyer n from rSTATIC-MATCHING under the modified instance $I^{(n,j)}$. We call $u_n^{(n,j)}$ a shadow utility. Note that the shadow utility captures the competition from other resources when buyer n considers purchasing resource j. The larger $u_n^{(n,j)}$ is, the more likely buyer n is to purchase resources other than j. Then we define the effective valuation of buyer n on resource j as

$$\hat{v}_{n,j} = v_{n,j} - \min\{u_n^{(n,j)}, v_{n,j}\}. \tag{D.2}$$

Since $u_n^{(n,j)} \geq 0$, we have $\hat{v}_{n,j} \in [0, v_{n,j}]$, which can be smaller than the lower bound $v_{\min,k}$. The shadow utility $u_n^{(n,j)}$ and effective valuation $\hat{v}_{n,j}$ satisfy the following properties.

(i) The utility of buyer n is not less than its shadow utility, i.e., $u_n \geq u_n^{(n,j)}$. If resource j has been sold out upon the arrival of buyer n, then $u_n = u_n^{(n,j)}$. Otherwise, before the arrival of buyer n, rSTATIC-MATCHING is exactly the same under both instance I and $I^{(n,j)}$. Under the instance I, buyer n has an additional resource j to choose from, and thus her utility must satisfy $u_n \geq u_n^{(n,j)}$.

(ii) Buyer n purchases resource j if her effective valuation exceeds the posted price p_j, i.e., $\hat{v}_{n,j} > p_j$, and the resource j has not been sold out. This is because $\hat{v}_{n,j} > p_j$ implies $v_{n,j} - p_j > u_n^{(n,j)}$, indicating that purchasing resource j yields a greater utility than all other resources.

Recall $\mathcal{N}_j^*$ contains buyers that purchase resource j in the offline algorithm. Let $(\hat{v}_{1,j}^*, \hat{v}_{2,j}^*, \ldots, \hat{v}_{N_j^*,j}^*)$ denote a sequence of non-decreasing effective valuations of buyers in $\mathcal{N}_j^*$, where $N_j^* = |\mathcal{N}_j^*|$. Furthermore, define a non-decreasing sequence of thresholds $(y_{0,j}, y_{1,j}, y_{2,j}, \ldots, y_{N_j^*,j})$ such that $y_{0,j} = 0$ and $\phi_j(y_{i,j}) = \hat{v}_{i,k}^*, \forall i \in [N_j^*]$. Let $\nu_j(y) = \sum_{n\in[N]} X_{n,j}$ denote the total number of sold copies of resource j by rSTATIC-MATCHING when the static price of resource j is given by $\phi_j(y)$. Then we have

$$\nu_j(y) = \sum\nolimits_{n\in[N]} X_{n,j} \geq \sum\nolimits_{i\in[N_j^*]} \mathbb{I}\{y \leq y_{i,j}\}, \forall y \in [0,1]. \tag{D.3}$$

To see the above inequality, if resource j is not sold out after the execution of the entire instance, then all buyers with effective valuations larger than $\phi_j(y)$ purchase resource j, and thus the inequality holds. If resource j is sold out, then $\nu_j(y) = C_k \geq \sum_{i\in[N_j^*]} \mathbb{I}\{y \leq y_{i,j}\}$.

Conditioned on the shadow utilities $\{u_i^{(i,j)}\}_{i\in[N_j^*]}$ from buyers in $\mathcal{N}_j^*$, the expected revenue earned from resource j can be lower bounded by

$$\mathbb{E}\Big[r_j\Big|\{u_i^{(i,j)}\}_{i\in[N_j^*]}\Big] \geq \int_0^1 \nu_j(y)\phi_j(y)\mathrm{d}y \tag{D.4}$$

$$= \sum_{i=1}^{N_j^*}\int_0^{y_{i,j}} \phi_j(y)\mathrm{d}y \tag{D.5}$$

$$\geq \sum_{i=1}^{N_j^*}\Big[\phi_j\big(y_{i,j}\big) - \Big(1-\tfrac{1}{\alpha_j}\Big)v_{i,j}\Big] \tag{D.6}$$

$$\geq \frac{1}{\alpha_j}\sum_{i=1}^{N_j^*} v_{i,j} - \sum_{i=1}^{N_j^*} u_i^{(i,k)}, \tag{D.7}$$

where inequality (D.5) is obtained by substituting equation (D.3) and exchanging the summations. To see the inequality (D.6), note that the designed inverse CDF function ϕ_j in equation (5.5) can ensure the following inequalities:

1. When $y_{i,j} \in [0, \omega_k)$, we have

$$\phi_j\big(y_{i,j}\big) - \int_0^{y_{i,j}} \phi_j(x)\mathrm{d}x \leq \Big(1-\frac{1}{\alpha_j}\Big)v_{\min,k} \leq \Big(1-\frac{1}{\alpha_j}\Big)v_{i,j},$$

   where $v_{i,j} \geq v_{\min,k}$ since buyer i is from set $\mathcal{N}_j^*$ and thus interested in resource j.

2. When $y_{i,j} \in [\omega_k, 1]$, we have

$$\begin{aligned}\phi_j\big(y_{i,j}\big) &\leq \int_0^{y_{i,j}} \phi_j(x)\mathrm{d}x + \Big(1-\tfrac{1}{\alpha_j}\Big)\phi_j\big(y_{i,j}\big)\\ &\leq \int_0^{y_{i,j}} \phi_j(x)\mathrm{d}x + \Big(1-\tfrac{1}{\alpha_j}\Big)v_{i,j},\end{aligned}$$

   where the last inequality holds since $v_{i,j} \geq \hat{v}_{i,j} = \phi_j(y_{i,j})$.

Based on inequality (D.1), we can have

$$\begin{aligned}\mathbb{E}[r_j] = \mathbb{E}[\mathbb{E}[r_j|\{u_i^{(i,j)}\}_{i\in[N_j^*]}]] &\geq \frac{1}{\alpha_j}\sum\nolimits_{i=1}^{N_j^*} v_{i,j} - \sum\nolimits_{i=1}^{N_j^*}\mathbb{E}[u_i^{(i,j)}]\\ &\geq \frac{1}{\alpha_j}\sum\nolimits_{i=1}^{N_j^*} v_{i,j} - \sum\nolimits_{i=1}^{N_j^*}\mathbb{E}[u_i],\end{aligned}$$

which completes the proof of inequality (D.1). □

D.2 Proof of Theorem 5.2

We consider a setup for `Matching-BW` with J resources, each resource having C identical copies. We furthermore assume that for each resource type j, the valuation v_j lies in the interval $[L, U]$ for all resource types; that is, the range over which the valuation for each resource type fluctuates is identical and equal to $[L, U]$. Let π be a permutation on the set $\{1, 2, \ldots, J\}$. Let $\mathcal{A}_l(C, v)$, $l \in \{1, 2, \ldots, J\}$, denote a batch of C identical buyers with the same valuation v (where $v \in [L, U]$) and interest in resources from $\pi(l)$ to $\pi(J)$. We construct the hard instances as follows.

Stage I. The instance in this stage is the classic upper-triangle instance from the online matching literature [DJ12]. In particular, we consider an instance consisting of J batches of buyers in the form of $\mathcal{I}_L = \mathcal{A}_1(C, L) \oplus \mathcal{A}_2(C, L) \oplus \cdots \oplus \mathcal{A}_J(C, L)$. The $(l+1)$-th batch of buyers is interested in the same resources that the l-th batch is interested in except for the resource $\pi(l)$. We claim that the optimal randomized online algorithm, denoted by `ALG`, is a balancing algorithm that equally assigns buyers in each batch to their interested resources in expectation. Specifically, the balancing algorithm `ALG` in expectation assigns $q_{i,l}$ buyers of the batch $\mathcal{A}_l(C, L)$ to resource $i \in [J]$, where

$$q_{i,l} = \begin{cases} \frac{C}{J-l+1} & \pi(i) \geq l, \\ 0 & \pi(i) < l. \end{cases}$$

In addition to balancing its inventory, the optimal randomized online algorithm `ALG` also needs to consider a protection threshold of ωC for accepting buyers of valuation L such that once the algorithm allocates one resource up to the protection threshold, it would stop allocating from that resource. Thus, the final utilization of resource i is $\min\{\omega C, \sum_{l \in [J]} q_{i,l}\}$ under $\mathcal{I}_L$. The value of ω will be determined after considering instances in Stage II in such a way that the algorithm achieves the best competitive ratio.

The online balancing algorithm `ALG` is optimal for the instances in Stage I. This is because any other algorithm with an imbalanced assignment of buyers to resources in each batch will perform worse than the online balancing algorithm `ALG` under some permutation π. For instance, π can be chosen by the adversary in such a way that the resource with the lowest utilization level is discarded in the following batch. Since the online balancing algorithm is indifferent to the permutation π, we consider w.l.o.g. that π is an identity permutation hereinafter.

In contrast to the online balancing algorithm `ALG`, the offline algorithm assigns all buyers in batch $\mathcal{A}_l(C, L)$ to resource $\pi(l)$, and can attain $\texttt{OPT}(\mathcal{I}_L) = LCJ$. Thus, an online balancing algorithm `ALG` with threshold ω can achieve

$$\begin{aligned}\texttt{ALG}(\omega, \mathcal{I}_L) &= L\sum\nolimits_{i=1}^{J} \min\left\{\omega C, \sum\nolimits_{l=1}^{i} \frac{C}{J-l+1}\right\} \\ &= L\sum\nolimits_{i=1}^{l_\omega}\sum\nolimits_{l=1}^{i} \frac{C}{J-l+1} + L\omega C(J-l_\omega) \\ &\approx LCl_\omega = LCJ(1-e^{-\omega}), \end{aligned} \tag{D.8}$$

where l_ω is the index such that $\sum_{l=1}^{l_\omega}\frac{C}{J-l+1} = \omega C$. Thus $\omega \approx \ln\frac{J}{J-l_\omega}$, and $l_\omega \approx J(1-e^{-\omega})$ as $J\to\infty$. The last equation is obtained by observing that $\sum_{i=1}^{l_\omega}\sum_{l=1}^{i}\frac{1}{J-l+1} = \sum_{l=1}^{l_\omega}\sum_{i=l}^{l_\omega}\frac{1}{J-l+1} = l_\omega - (J-l_\omega)\sum_{l=1}^{l_\omega}\frac{1}{J-l+1} = l_\omega - (J-l_\omega)\omega$.

Stage II. At the end of Stage I, the resources from $\pi\big(J(1-e^{\omega})\big)$ to $\pi(J)$ have sold ωC units of resources. The arrivals of Stage II are constructed similarly to the worst-case instance of the `kSelection-cost` in Chapter 2 in the proof of lower-bound obtain for this problem, and only interested in one of the last Je^{ω} resources. Specifically, an instance of Stage II with maximum value v ($v\in(L,U]$) is denoted by $\mathcal{I}_v = \mathcal{I}_L \oplus \mathcal{A}_J(C, L+\epsilon)\oplus\mathcal{A}_J(C, L+2\epsilon)\oplus\cdots\oplus\mathcal{A}_J(C,v)$, which consists of a sequence of batches that are only interested in the resource $\pi(J)$ and their valuations continuously increase from L to v. Then the offline algorithm under instance $\mathcal{I}_v$ assigns batch $\mathcal{A}_l(C,L)$ to resource $\pi(l)$ for $l\in[J-1]$ and assigns the batch $\mathcal{A}_J(C,v)$ to resource $\pi(J)$, achieving the optimal social welfare $\texttt{OPT}(\mathcal{I}_v) = LC(J-1)+vC$. Let $g(v)$ denote the average representative function of the resource J by running a randomized online algorithm under instance $\mathcal{I}_v$, where $g(L)=\omega C$ is the maximum sold units in Stage I. The expected social welfare of the online algorithm characterized by g can be described as $\texttt{ALG}(g,\mathcal{I}_v) = LCl_\omega + \int_L^v u dg(u)$, where the first term is return from the instance in Stage I and the second term is return in Stage II. Any α-competitive online algorithm must satisfy $\texttt{ALG}(g,\mathcal{I}_v)\geq\frac{1}{\alpha}\texttt{OPT}(\mathcal{I}_v), \forall v\in[L,U]$. This gives

$$LCl_\omega \geq \frac{1}{\alpha}LCJ, \tag{D.9}$$

$$LCl_\omega + \int_L^v u dg(u) \geq \frac{1}{\alpha}\left[LC(J-1)+vC\right], \quad \forall v\in(L,U]. \tag{D.10}$$

Equation (D.9) gives us a lower bound of the competitive ratio $\alpha\geq\frac{e^{\omega}}{e^{\omega}-1}$, and equation (D.10) gives

$$g(v) \geq \frac{C}{\alpha} + \frac{1}{v}\int_L^v g(u)du + \frac{L\omega C}{v} - \frac{LC}{v\alpha} \geq \omega C + \frac{C}{\alpha}\ln\frac{v}{L}, \quad \forall v\in(L,U].$$

Since $g(U)\leq C$, we have $\alpha\geq\frac{\ln\theta}{1-\omega}$, where $\theta=\frac{U}{L}$. Thus, the lower bound of `Matching-BW` is

$$\alpha \geq \min_{\omega\in[0,1]}\max\left\{\frac{e^{\omega}}{e^{\omega}-1}, \frac{\ln\theta}{1-\omega}\right\},$$

which is achieved when the threshold ω is the solution of $\frac{e^\omega}{e^\omega - 1} = \frac{\ln \theta}{1-\omega}$. $\square$

Appendix E

Proofs of Chapter 6

E.1 Proof of Proposition 6.1

Proof. We employ a *randomized* version of the LP-free certificate approach (used to prove Proposition 4.1) to derive an upper bound on the competitive ratio of Algorithm 6 for any instance $I \in \mathcal{I}^{\texttt{Matching-RR}}$. In this randomized variant, the set of linear constraints defining the certificate system to prove the algorithm's competitiveness need to be satisfied only in expectation.

In the randomized version, depending on the value of the random seed vector $\mathbf{s}$, we first define the set of variables $\{u_n(\mathbf{s}), \lambda_{n,j}^{(i)}(\mathbf{s}), \theta_j^{(i)}(\mathbf{s})\}_{i\in[k],n\in[N]}$. Then, the final set of variables forming the system of linear constraints is given by

$$u_n = \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}[u_n(\mathbf{s})] \quad \text{and} \quad \theta_j^{(i)} = \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}[\theta_j^{(i)}(\mathbf{s})] = \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}\Big[\sum_{n\in[N]} \lambda_{n,j}^{(i)}(\mathbf{s})\Big].$$

The system of constraints is similar to that in the proof of Proposition 4.1, namely:

$$\sum_{n\in[N]} u_n + \sum_{j\in[J]} \sum_{i\in[C_j]} \theta_j^{(i)} \le G \cdot \texttt{ALG}(I), \tag{E.1}$$

$$\theta_j^{(i)} + \sum_{n\in P_j^{(i)}} u_n \ge \sum_{n\in P_j^{(i)}} d_n \cdot \rho_{n,j} = \texttt{OPT}_j^{(i)}, \quad \forall j \in [J],\ \forall i \in [C_j], \tag{E.2}$$

where

$$G = \max_{j\in[J]} \left\{ \left(1 + \ln\Big(\frac{\rho_{\max,j}}{\rho_{\min,j}}\Big)\right) \cdot \alpha_j \right\},$$

the set $P_j^{(i)}$ is the set of buyers to whom the offline clairvoyant algorithm allocates the i-th unit of resource j, and

$$\mathtt{OPT}_j^{(i)} = \sum_{n \in P_j^{(i)}} \rho_{n,j} d_{n,j}.$$

If we can find a solution to the above system of linear constraints, the G-competitiveness of Algorithm 6 follows.

Let $x_n(\mathbf{s})$ denote the decision variable determined by Algorithm 6 for each buyer n; this is a random variable that depends on the random seed vector $\mathbf{s}$. It can be verified that, according to Algorithm 6, once the random seed vector $\mathbf{s}$ is realized, the value of $x_n(\mathbf{s})$ is determined deterministically.

Let us define the set of variables $\{u_n(\mathbf{s}), \lambda_{n,j}^{(i)}(\mathbf{s}), \theta_j^{(i)}(\mathbf{s})\}_{i \in [C_j], j \in [J], n \in [N]}$ as follows:

$$u_n(\mathbf{s}) = \frac{G}{3} \cdot \rho_{n,j_n^*(\mathbf{s})} \cdot d_{n,j_n^*(\mathbf{s})} \cdot x_n(\mathbf{s}), \tag{E.3}$$

$$\begin{aligned} \lambda_{n,j}^{(i)}(\mathbf{s}) = & \sum_{\tau \in B_{n,j}^{(i)}(\mathbf{s})} \frac{G}{3} \cdot \big(a_{n+1} - a_n\big) \cdot \phi_{2,j}(\mathbf{s}_j) \cdot x_\tau(\mathbf{s}) + \\ & \sum_{\tau \in C_{n,j}^{(i)}(\mathbf{s})} \frac{G}{3} \cdot \big(2 \cdot d_{\tau,j} + a_\tau - a_n\big) \cdot \phi_{2,j}(\mathbf{s}_j) \cdot x_\tau(\mathbf{s}), \end{aligned} \tag{E.4}$$

where

$$B_{n,j}^{(i)}(\mathbf{s}) = \Big\{\tau \in [1,n] \Big| j_\tau^*(\mathbf{s}) = j, i_{\tau,j}^*(\mathbf{s}) = i, a_\tau + d_{\tau,j} > a_{n+1}\Big\},$$

and

$$C_{n,j}^{(i)}(\mathbf{s}) = \Big\{\tau \in [1,n] \Big| j_\tau^*(\mathbf{s}) = j, i_{\tau,j}^*(\mathbf{s}) = i, a_n < a_\tau + d_\tau < a_{n+1}\Big\}.$$

Furthermore, we set

$$\theta_j^{(i)}(\mathbf{s}) = \sum_{n \in [N]} \lambda_{n,j}^{(i)}(\mathbf{s}).$$

Feasibility of First Constraint Eq. (E.1)**.** Let $\mathtt{ALG}(\mathbf{s})$ denote the expected objective value of Algorithm 6 on instance I, conditioned on a given realization of the random seed vector $\mathbf{s}$. The expectation is taken over all sources of randomness *other than* that introduced by $\mathbf{s}$. In addition to the randomness from the seed vector $\mathbf{s}$, the algorithm employs other random processes (e.g., in Line 17). Consequently, $\mathtt{ALG}(\mathbf{s})$ represents the expected objective value of the algorithm with respect to these additional random processes.

We wish to show the following stronger version of Eq. (E.1):

$$\sum_{n\in[N]} u_n(\mathbf{s}) + \sum_{j\in[J]}\sum_{i\in[C_j]} \theta_j^{(i)}(\mathbf{s}) \le G\cdot \texttt{ALG}(\mathbf{s}). \tag{E.5}$$

where in above Eq. (E.1) holds on any realized value of the random seed vector $\mathbf{s}$.

To begin, let $\Delta_n^{\texttt{ALG}(\mathbf{s})}$ be the increase in the expected objective value of Algorithm 6 after processing buyer n. By reasoning analogous to that in Proposition 4.1, we have

$$\Delta_n^{\texttt{ALG}(\mathbf{s})} = \rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}),$$

and thus

$$\texttt{ALG}(\mathbf{s}) = \sum_{n\in[N]} \Delta_n^{\texttt{ALG}(\mathbf{s})} = \sum_{n\in[N]} \rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}).$$

Next, consider the left-hand side of Eq. (E.2):

$$\begin{aligned}
&\sum_{n\in[N]} u_n(\mathbf{s}) + \sum_{j\in[J]}\sum_{i\in[C_j]} \theta_j^{(i)}(\mathbf{s}) \\
=& \sum_{n\in[N]} \frac{G}{3}\rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}) + \sum_{j\in[J]}\sum_{i\in[C_j]}\sum_{n\in[N]} \lambda_{n,j}^{(i)}(\mathbf{s}) \\
\le& \sum_{n\in[N]} \frac{G}{3}\rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}) + \sum_{j\in[J]}\sum_{i\in[C_j]} \phi_{2,j}(\mathbf{s}_j) \sum_{n\in[N]} x_n(\mathbf{s}) \mathbf{1}_{\{j_n^*(\mathbf{s})=j, i^*_{j_n^*(\mathbf{s})}=i\}} \\
&\quad \cdot \Big[\sum_{n'\in D_{n,j}^{(i)}} \tfrac{G}{3}\big(a_{n'} - a_{n'-1}\big) + \tfrac{G}{3}\big(2d_{n,j} + a_n - a_{n_j^*}\big)\Big] \\
\le& \sum_{n\in[N]} \frac{G}{3}\rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}) + \sum_{j\in[J]}\sum_{i\in[C_j]}\sum_{n\in[N]} x_n(\mathbf{s}) \mathbf{1}_{\{j_n^*(\mathbf{s})=j, i^*_{\nu^*_{n,j}(\mathbf{s})}=i\}} \\
&\quad \cdot \Big[2\cdot \tfrac{G}{3}\rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})}\Big] \\
=& \sum_{n\in[N]} 3\tfrac{G}{3}\rho_{n,j_n^*(\mathbf{s})} d_{n,j_n^*(\mathbf{s})} x_n(\mathbf{s}) \\
\le& G\texttt{ALG}(\mathbf{s}).
\end{aligned}$$

In the above derivation, $D_{n,j}^{(i)} = \big\{n' \in \{n, n+1, \ldots, N\} : a_n + d_{n,j} > a_{n'+1}\big\}$ and

$$\nu^*_{n,j} = \max\big\{n' \ge n : a_{n'} < a_n + d_{n,j}\big\}.$$

The first inequality in above follows from rearranging the terms in Eq. (E.4). The second inequality follows from definition of the set $D^{(i)}_{n,j}$ and $\nu^*_{n,j}$. Therefore, combining all these pieces, the inequality in Eq. (E.5) follows.

Feasibility of the Second Set of Constraints (E.2)**.** Using similar reasoning as one done in Proposition 4.1 to prove feasibility of set of constraints in Eq. (C.6), it suffices to show that, for each buyer $n \in P^{(i)}_j$,

$$\sum_{n'=n}^{N} \lambda^{(i)}_{n',j} \mathbf{1}_{\left\{a_n + d_{n,j} > a_{n'}\right\}} + u_n \geq d_{n,j}\rho_{n,j}.$$

Using a similar argument as done in proof of Proposition 4.1 to lower-bound the term $\sum_{n'=n}^{N} \lambda^{(i)}_{n'} \mathbf{1}_{\{a_n + d_n > a_{n'}\}}$ done in Subsection C.7, one can show that we have

$$\begin{aligned}
&\sum_{\tau=n}^{N} \lambda^{(i)}_{\tau,j}(\mathbf{s}) \mathbf{1}_{\{a_n + d_{n,j} > a_\tau\}} \\
&\geq \phi_{2,j}(\mathbf{s}_j) \left[\frac{2G}{3} z'_1 \phi_{1,j}(z'_1) - \frac{G}{3} z'_L \phi_{1,j}(z'_1) + \int_{z'_1}^{z'_L} \frac{2G}{3} \phi_{1,j}(\eta) d\eta \right. \\
&\qquad\qquad \left. + \frac{G}{3} d_{n,j} \max\{0, y_{n,j}(\mathbf{s}) - z'_L\} \right],
\end{aligned} \tag{E.6}$$

where $z'_1 \in [0, z'_L]$ and $z'_L \in \left[0, \phi^{-1}_{1,j}(d_{n,j})\right]$.

Note: The set of variables $\{\lambda^{(i)}_{n,j}\}_{n\in[N]}$ is defined in Eq. (E.4) for each unit i of each resource type j. We could have updated these variables following a similar scheme as in the proof of Proposition 4.1, where once $j^*_n(\mathbf{s}) = j$ and $i^*_{n,j} = i$ the subset of variable $\{\lambda^{(i)}_{n,j}\}_{n\in[N]}$ get updated. However, the final value of each variable would be identical to that given in Eq. (E.4).

Furthermore, we can lower-bound the variable $u_n(\mathbf{s})$ under the condition that $s_j \leq \phi^{-1}_{2,j}(\rho_{n,j})$ as follows:

$$\begin{aligned}
u_n(\mathbf{s}) &= \frac{G}{3} \cdot \rho_{n,j^*_n(\mathbf{s})} \cdot d_{n,j^*_n(\mathbf{s})} \cdot x_n(\mathbf{s}) \\
&\geq \frac{G}{3} \cdot \rho_{n,j} \cdot d_{n,j} \cdot x_{n,j} \\
&\geq \frac{G}{3} \cdot \phi_{2,j}(s_j) \cdot d_{n,j} \cdot \max\Big\{0, \phi^{-1}_{1,j}(d_{n,j}) - y_{n,j}(\mathbf{s})\Big\}, \quad \textit{provided that } s_j \leq \phi^{-1}_{2,j}(\rho_{n,j}).
\end{aligned}$$

In the above, the first inequality follows from the fact that $j \in R$ (since $s_j \leq \phi^{-1}_{2,j}(\rho_{n,j})$) and $j^*_n(\mathbf{s}) = \arg\max_{j\in R} x_{n,j}\rho_{n,j}d_{n,j}$, and the second inequality is a consequence of the definition of $x_{n,j}$

along with the condition $s_j \leq \phi_{2,j}^{-1}(\rho_{n,j})$.

Putting together the lower bounds for $\sum_{\tau=n}^{N} \lambda_{\tau,j}^{(i)}(\mathbf{s})\mathbf{1}_{\{a_n+d_{n,j}>a_\tau\}}$ and for the variable $u_n(\mathbf{s})$, it follows that

$$\begin{aligned}
&\sum_{\tau=n}^{N} \lambda_{\tau,j}^{(i)}(\mathbf{s})\mathbf{1}_{\{a_n+d_{n,j}>a_\tau\}} + u_n(\mathbf{s}) \\
&\geq \phi_{2,j}(s_j)\Bigg[\frac{2G}{3}z_1'\phi_{1,j}(z_1') - \frac{G}{3}z_L'\phi_{1,j}(z_1') + \int_{z_1'}^{z_L'} \frac{2G}{3}\phi_{1,j}(\eta)d\eta \\
&\quad + \frac{G}{3}d_{n,j}\max\Big\{0, y_{n,j}(\mathbf{s}) - z_L'\Big\} + \frac{G}{3}d_{n,j}\max\Big\{0, \phi_{1,j}^{-1}(d_{n,j}) - y_{n,j}(\mathbf{s})\Big\}\Bigg],
\end{aligned} \tag{E.7}$$

provided that $s_j \leq \phi_{2,j}^{-1}(\rho_{n,j})$, and for some

$$z_1' \in [0, z_L'] \quad \text{and} \quad z_L' \in \Big[0, \phi_{1,j}^{-1}(d_{n,j})\Big].$$

Arguing as before, and using the fact that $\phi_{1,j}$ satisfies the constraints (6.2) and (6.3), it follows that the right-hand side of (E.7) is at least $\frac{G}{\alpha_j}\phi_{2,j}(\mathbf{s}_j)d_{n,j}$. Hence,

$$\sum_{\tau=n}^{N} \lambda_{\tau,j}^{(i)}(\mathbf{s})\mathbf{1}_{\left\{a_n+d_{n,j}>a_\tau\right\}} + u_n(\mathbf{s}) \geq \frac{G}{\alpha_j}\phi_{2,j}(\mathbf{s}_j)d_{n,j}, \qquad s_j \leq \phi_{2,j}^{-1}(\rho_{n,j}) \tag{E.8}$$

Taking the expectation over the randomness in $\mathbf{s}$, we have

$$\begin{aligned}
u_n + \sum_{\tau=n}^{N} \lambda_{\tau,j}^{(i)}\mathbf{1}_{\{a_n+d_{n,j}>a_\tau\}} &= \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}\big[u_n(\mathbf{s})\big] + \sum_{\tau=n}^{N} \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}\big[\lambda_{\tau,j}^{(i)}(\mathbf{s})\big] \\
&= \mathbb{E}_{\mathbf{s}\sim\mathcal{U}^J(0,1)}\Big[u_n(\mathbf{s}) + \sum_{\tau=n}^{N} \lambda_{\tau,j}^{(i)}(\mathbf{s})\mathbf{1}_{\{a_n+d_{n,j}>a_\tau\}}\Big] \\
&\geq \mathbb{E}_{\mathbf{s}_j\sim\mathcal{U}(0,1)}\Big[\frac{G}{\alpha_j}\phi_{2,j}(s_j)d_{n,j}\Big| s_j \leq \phi_{2,j}^{-1}(\rho_{n,j})\Big] \\
&= \int_{\eta=0}^{\phi_{2,j}^{-1}(\rho_{n,j})} \frac{G}{\alpha_j}\phi_{2,j}(\eta)d_{n,j}d\eta \\
&\geq \rho_{n,j}d_{n,j},
\end{aligned}$$

where the final inequality follows from the definition (design) of $\phi_{2,j}$ and the value of G defined in above. Thus, the second set of constraints in Eq. (E.2) holds. □

E.2 Proof of Proposition 6.2

In order to prove the lemma, we can construct a set of hard instances for the `Matching-RR` problem in a manner similar to [SJTB24]. Let

$$j^* = \arg\max_{j\in[J]} \frac{d_{\max,j}\rho_{\max,j}}{d_{\min,j}\rho_{\min,j}}.$$

Let $\mathcal{A}(k,\rho,d)$ denote a batch of k identical buyers, each with a per-unit-time valuation ρ (with $\rho \in [\rho_{\min,j^*}, \rho_{\max,j^*}]$) and a rental duration request d (with $d \in [d_{\min,j^*}, d_{\max,j^*}]$). Divide the uncertainty range with respect to buyers' valuations,

$$[d_{\min,j^*}\rho_{\min,j^*}, d_{\max,j^*}\rho_{\max,j^*}],$$

into $m-1$ subranges of equal length

$$\Delta_v = \frac{d_{\max,j^*}\rho_{\max,j^*} - d_{\min,j^*}\rho_{\min,j^*}}{m-1}.$$

Define

$$\mathcal{V} := \{v_i\}_{i\in[m]}, \quad \text{with} \quad v_i = d_{\min,j^*}\rho_{\min,j^*} + (i-1)\Delta_v, \quad \text{for } i \in [m].$$

For each $i \in [m]$, define the instance

$$\begin{aligned} I_{v_i} := \mathcal{A}\Big(k, \max\Big\{\frac{v_1}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_1}{\rho_{\min,j^*}}\Big\}\Big) \oplus \\ \mathcal{A}\Big(k, \max\Big\{\frac{v_2}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_2}{\rho_{\min,j^*}}\Big\}\Big) \oplus \\ \cdots \oplus \mathcal{A}\Big(k, \max\Big\{\frac{v_i}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_i}{\rho_{\min,j^*}}\Big\}\Big). \end{aligned}$$

This instance consists of a sequence of buyer batches, where the valuation of rental requests increases across batches. Specifically, the batch $\mathcal{A}\Big(k, \max\Big\{\frac{v_i}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_i}{\rho_{\min,j^*}}\Big\}\Big)$ is followed by the batch $\mathcal{A}\Big(k, \max\Big\{\frac{v_{i+1}}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_{i+1}}{\rho_{\min,j^*}}\Big\}\Big)$. Each buyer in the batch $\mathcal{A}\Big(k, \max\Big\{\frac{v_i}{d_{\max,j^*}}, \rho_{\min,j^*}\Big\}, \min\Big\{d_{\max,j^*}, \frac{v_i}{\rho_{\min,j^*}}\Big\}\Big)$ has a valuation equal to v_i, and their rental durations and per-unit-time valuations fall within the bounded ranges assumed in the original problem setting for resource type j^*. In this construction, all buyers—from the first batch with valuation v_1 to the last batch with valuation v_m—arrive within a short time interval $[0,\epsilon]$, where ϵ is a small constant satisfying $\epsilon < d_{\min,j^*}$.

The collection $\{I_{v_i}\}_{i\in[m]}$ is then considered as the set of hard instances for the `kRental-variable` problem. Since buyers in these instances arrive in a very short time interval, if a unit of a resource is allocated to a buyer in any batch that resource cannot be used by any other buyer in this instance.

Following the same reasoning as proof of Lemma 2.3 in [SJTB24], one can show that every online algorithm must have a competitive ratio of at least

$$\max_{j\in[J]} \left\{ 1 + \ln\left(\frac{d_{\max,j}}{d_{\min,j}}\right) + \ln\left(\frac{\rho_{\max,j}}{\rho_{\min,j}}\right) \right\}.$$