Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack*

Jiashuo Jiang† Will Ma‡ Jiawei Zhang §

Abstract

Prophet inequalities are a useful tool for designing online allocation procedures and comparing their performance to the optimal offline allocation. In the basic setting of k-unit prophet inequalities, the procedure of Alaei [2] with its celebrated performance guarantee of $1 - \frac{1}{\sqrt{k+3}}$ has found widespread adoption in mechanism design and general online allocation problems in online advertising, healthcare scheduling, and revenue management. Despite being commonly used for implementing a fractional allocation in an online fashion, the tightness of Alaei's procedure for a given k has remained unknown. In this paper we resolve this question, characterizing the tight bound by identifying the structure of the optimal online implementation, and consequently improving the best-known guarantee for k-unit prophet inequalities for all $k > 1$.

We also consider the more general online stochastic knapsack problem where each individual allocation can consume an arbitrary fraction of the initial capacity. Here we introduce a new "best-fit" procedure for implementing a fractionally-feasible knapsack solution online, with a performance guarantee of $\frac{1}{3+e^{-2}} \approx 0.319$, which we also show is tight with respect to the standard LP relaxation. This improves the previously best-known guarantee of 0.2 for online knapsack. Our analysis differs from existing ones by eschewing the need to split items into "large" or "small" based on capacity consumption, using instead an invariant for the overall utilization on different sample paths.

All in all, our results imply *tight* (non-greedy) Online Contention Resolution Schemes for k-uniform matroids and the knapsack polytope, respectively, which has further implications.

1 Introduction

Online stochastic knapsack and k-unit prophet inequalities are fundamental problems in online algorithms, often applied as crucial subroutines in mechanism design and general online resource allocation. In these problems, there is a single resource with an infinitely-divisible capacity normalized to 1. A sequence of queries is observed, each with a reward and a size. Upon observation, a query must be immediately either served, claiming its reward and spending capacity equal to its size; or rejected, which is the only option if the remaining capacity is less than its size. The (size, reward) pair for each query is initially unknown, but drawn independently from non-identical distributions that are given in advance. The objective is to maximize the total reward collected in expectation. The special case of k-unit prophet inequalities restrains all query sizes to be $1/k$.

Typically, the expected reward collected by the online algorithm is compared to that of a *prophet*, who sees the size and reward realizations in advance and can always make an optimal allocation of capacity in hindsight. The classical result is that in 1-unit prophet inequalities, where serving any query consumes the entire capacity, an online algorithm can collect at least 1/2 times the reward of the prophet. In general k-unit prophet inequalities, the best-known guarantees come from comparing the online algorithm to a *Linear Programming (LP) relaxation* of the prophet, which only has to satisfy the capacity constraint in expectation. In a celebrated result, Alaei [2] showed that an online algorithm can collect a factor of this LP relaxation which is lower-bounded by $1 - \frac{1}{\sqrt{k+3}}$, implying a guarantee of $1 - \frac{1}{\sqrt{k+3}}$ for k-unit prophet inequalities.

Another benefit of the LP relaxation approach is that it extends to highly general online allocation problems, with multiple resources and customer choice, modeling applications in online advertising [3], healthcare scheduling [26, 27], and revenue management [17]. Since these problems suffer from the curse of dimensionality, the aforementioned papers design efficient algorithms by first using a centralized LP to effectively "assign" each query to a resource. The decision of whether to serve an assigned query is then separately considered at the

*The full version of the paper can be accessed at `https://arxiv.org/abs/2107.02058`.

†Department of Technology, Operations & Statistics, Stern School of Business, New York University.
‡Decision, Risk, and Operations Division, Graduate School of Business, Columbia University.
§Department of Technology, Operations & Statistics, Stern School of Business, New York University.

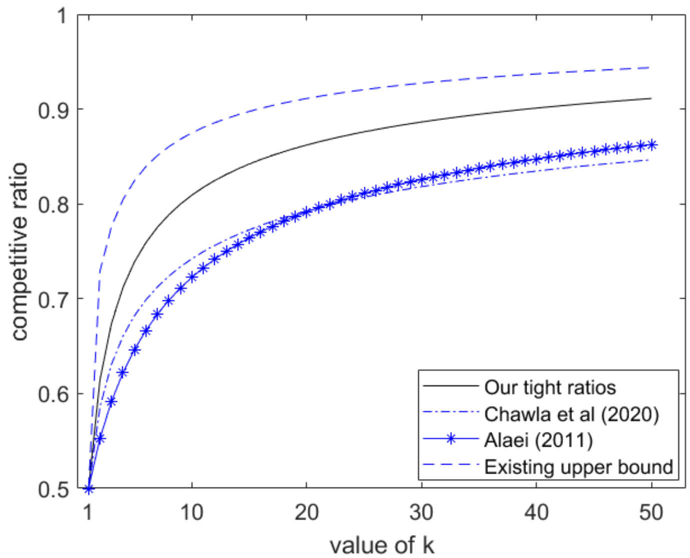


Figure 1: Our tight ratios in comparison to the previously best-known guarantees from [2, 10]. The previously best-known upper bounds were $1/2$ for $k = 1$ and $1 - e^{-k}k^k/k!$ for $k > 1$, with the latter inherited from the "correlation gap" in the i.i.d. special case [see 28].

resource level, using an algorithm for k-unit prophet inequalities or online knapsack. For the overall algorithm, its approximation ratio can be shown to reduce to the single resource with the *worst* guarantee in the reward that an online algorithm can collect, relative to a fractionally-feasible allocation of that resource coming from the LP relaxation. (This reduction is formally explained in the full version [19].)

For these reasons, bounding the performance of online algorithms relative to a fractional allocation of a single resource is an important question. In this vein, if the resource is always consumed in increments of $1/k$, then the best-known lower bound is the aforementioned $1 - \frac{1}{\sqrt{k+3}}$ of [2], which has been recently improved[1] for small values of k by [10]. On the other hand, if the resource can be consumed in arbitrary increments like in the online knapsack problem, then the best-known guarantee is 0.2 due to [13].

In this paper we improve the guarantees for both of these problems, and moreover show that our improvements are *tight* relative to the fractional relaxation. More specifically, we improve k-unit prophet inequalities (see Figure 1 and Table 1) for every integer $k > 1$, impacting not only the aforementioned papers, but also other papers in mechanism design [5, 6] and online assortment optimization [14, 24] in which the guarantee of $1 - \frac{1}{\sqrt{k+3}}$ is referenced. For online knapsack, we improve the best-known guarantee from 0.2 to $\frac{1}{3+e^{-2}} \approx 0.319$. The fact that our improvements are tight demonstrates the *limits* of any argument that holds relative to the fractional allocation when it comes to proving bounds for either single-resource or multi-resource online allocation.

Table 1: Our tight ratios for multi-unit prophet inequalities. The previous lower bounds are obtained as the maximum of the ratios in [2, 10]. The previously best-known upper bounds were $1/2$ for $k = 1$ and $1 - e^{-k}k^k/k!$ for $k > 1$, with the latter inherited from the "correlation gap" in the i.i.d. special case [see 28].

value of k	1	2	3	4	5	6	7	8
Existing lower bounds	0.5000	0.5859	0.6309	0.6605	0.6821	0.6989	0.7125	0.7240
Our tight ratios	0.5000	0.6148	0.6741	0.7120	0.7389	0.7593	0.7754	0.7887
Existing upper bounds	0.5000	0.7293	0.7760	0.8046	0.8245	0.8394	0.8510	0.8604

[1]To be precise, the bounds in [10] are stated relative to the prophet. However, it is easily verified that their arguments also hold relative to the stronger LP benchmark, and hence cannot improve beyond the tight guarantees given in this paper.

1.1 New techniques for k-unit prophet inequalities. We first formalize the problem of comparing an online algorithm to a fractional allocation. The fractional allocation is described by a vector $\boldsymbol{p} = (p_1, \ldots, p_T)$, with p_t denoting the probability with which query t should be served, which is fractionally-feasible in that $\sum_t p_t \le k$, since at most k queries can be served in the k-unit prophet inequality problem. The appropriate values of p_t are obtained by solving the LP relaxation. Given these values, our problem of interest treats each query t as having a binary state—*active* if its realized reward is observed to lie in its top p_t'th quantile; and inactive otherwise. It has been shown that if an online algorithm can serve every query with probability γ conditional on it being active, then the algorithm has a guarantee of γ relative to the LP relaxation as well as the prophet—we formally prove these facts in Section 3. We now proceed with this reduced problem, formalized in Definition 1.1 below, which we will call an *Online Contention Resolution Scheme (OCRS)* (we elaborate on the OCRS literature in Section 2).

Definition 1.1. (k-unit OCRS Problem) *There is a sequence of queries $t = 1, \ldots, T$, each of which is active independently according to a known probability p_t. Whether a query is active is sequentially observed, and active queries can be immediately served or rejected, while inactive queries must be rejected. At most k queries can be served in total, and it is promised that $\sum_t p_t \le k$. The goal of an online algorithm is to serve every query t with probability at least γ conditional on it being active, for a constant $\gamma \in [0, 1]$ as large as possible, potentially with the aid of randomization.*

It is easy to see[2] that despite $\boldsymbol{p}$ being fractionally-feasible, a guarantee of $\gamma = 1$ in Definition 1.1 is generally impossible. The work of [2] implies a solution to Definition 1.1 with $\gamma = 1 - \frac{1}{\sqrt{k+3}}$. Presented in the slightly different context of a "γ-Conservative Magician", Alaei's procedure has the further appealing property that it does not need to know vector $\boldsymbol{p}$ in advance, as long as each p_t is revealed when query t is observed, and it is promised that $\sum_t p_t \le k$. However, it has remained unknown whether Alaei's $\boldsymbol{p}$-agnostic procedure or its analyzed bound of $\gamma = 1 - \frac{1}{\sqrt{k+3}}$ is tight for an arbitrary positive integer k. In this paper we resolve this question, in the following steps.

1. Under the assumption that $\boldsymbol{p}$ is known, we formulate the optimal k-unit OCRS problem using a new LP. This LP tracks the probability distribution of the capacity utilization, which must lie in $\{0, \frac{1}{k}, \ldots, 1\}$, over time $t = 1, \ldots, T$. The decision variables correspond to subdividing and selecting sample paths at each time t, with total measure exactly γ, on which the algorithm will serve query t whenever it is active. This selection is constrained to sample paths with at least $\frac{1}{k}$ capacity remaining, which is enforced in the LP through tracking the capacity utilization. Finally, γ is also a decision variable, with the objective being to maximize γ.

2. For an arbitrary $\boldsymbol{p}$, we characterize an optimal solution to this LP based on the structure of its dual. The optimal selection prioritizes sample paths with the *least* capacity utilized, at every time t, *irrespective* of the values $p_{t+1}, p_{t+2}, \ldots$ in the future. Such a solution corresponds to the γ-Conservative Magician from [2], except that γ, instead of being fixed to $1 - \frac{1}{\sqrt{k+3}}$, is set to an optimal value which depends on the vector $\boldsymbol{p}$.

3. We derive a closed-form expression for this optimal value of γ as a function of $\boldsymbol{p}$. We show that γ is minimized when $p_t = k/T$ for each t and $T \to \infty$, corresponding to a Poisson distribution of rate k. We characterize this infimum value of γ using an ODE and provide an efficient procedure for computing it numerically.

For any k, let γ_k^* denote the infimum value of γ described in Step 3 above. The conclusion is that setting $\gamma = \gamma_k^*$ is a feasible solution to Definition 1.1, with $\gamma_k^* > 1 - \frac{1}{\sqrt{k+3}}$ for all $k > 1$, achieved using the $\boldsymbol{p}$-agnostic procedure described above. Moreover, the guarantee of γ_k^* is best-possible, since even a procedure that knows $\boldsymbol{p}$ in advance cannot do better than a γ-Conservative Magician with an optimized value of γ, which in the Poisson worst case can be as low as γ_k^*.

Comparison to [3]. k-unit prophet inequalities have been analyzed using LP's before in [3], who formulate a primal LP encoding the adversary's problem of minimizing an online algorithm's optimal dynamic programming

[2]For example, suppose $k = 1$, $T = 2$, and $p_1 = p_2 = 1/2$. If we attempt to set $\gamma = 1$, then the first query would be served ex-ante w.p. $1/2$, i.e. whenever it is active. This means that half the time no capacity would remain for query 2, i.e. half the time query 2 is active it does not get served. For this example, the optimal value of γ can be calculated to be $2/3$.

value. They then use an auxiliary "Magician's problem", analyzed through a "sand/barrier" process, to construct a feasible dual solution with $\gamma = 1 - \frac{1}{\sqrt{k+3}}$. By contrast, we directly formulate the k-unit OCRS problem using an LP under the assumption that the vector $\boldsymbol{p}$ is known. Our LP dual along with complementary slackness allows us to establish the structure of the optimal k-unit OCRS, showing that it indeed corresponds to a γ-Conservative Magician. However, in our case γ is set to a value dependent on $\boldsymbol{p}$, which we show is always at least γ_k^*, and strictly greater than $1 - \frac{1}{\sqrt{k+3}}$ for all $k > 1$.

Comparison to [27]. The values of γ_k^* we derive have previously appeared in [27] through the stochastic analysis of a "reflecting" Poisson process. Our work differs by establishing *optimality* for these values γ_k^*, as the solutions to a sequence of optimization problems from our framework. Moreover, their paper assumes Poisson arrivals to begin with, while we allow arbitrary probability vectors $\boldsymbol{p}$ and show the limiting Poisson case to be the worst case.

Additional upper bounds for prophet inequalities. Through our LP's and complementary slackness, we can convert the Poisson worst case for the k-unit OCRS problem into an explicit instance of k-unit prophet inequalities, on which the reward of any online algorithm relative to the LP relaxation is upper-bounded by γ_k^*. Moreover, by modifying such an instance, we also provide a new upper bound of 0.6269 relative the prophet when $k = 2$ (Proposition 4.1). To the best of our knowledge, this is the first upper bound for $k > 1$ relative to the *prophet* to appear in the literature. Since our lower bound of γ_2^* is approximately 0.6148, it shows that not much improvement is possible even if one were to compare against the prophet instead of the fractional relaxation.

1.2 New techniques for online knapsack. We formalize the problem of comparing an online algorithm to a fractional allocation in the more general knapsack setting, where for now we assume the sizes to be deterministic. Like before, p_t denotes the probability of query t being served in the fractional allocation, and this can be interpreted as an Online Contention Resolution Scheme problem for the knapsack polytope.

Definition 1.2. (Knapsack OCRS Problem) *There is a sequence of queries $t = 1, \ldots, T$, each with a known size $d_t \in [0, 1]$ and a known probability p_t with which it will be active independently. Whether a query is active is sequentially observed, and active queries can be immediately served or rejected, while inactive queries must be rejected. The total size of queries served cannot exceed 1, and it is promised that $\sum_t p_t d_t \le 1$. The goal of an online algorithm is to serve every query t with probability at least γ conditional on it being active, for a constant $\gamma \in [0, 1]$ as large as possible.*

Similar to our approach for k-unit prophet inequalities, we design OCRS's for knapsack by tracking the distribution of capacity utilization over time. We select for each query t a γ-measure of sample paths on which it should be served whenever active, under the constraint that these paths have current utilization no more than $1 - d_t$. Note that in the knapsack problem, we need to always maintain a γ-measure of sample paths on which utilization is 0, in case a query T with $d_T = 1$ and $p_T = \varepsilon$ arrives at the end. Accordingly, in stark contrast to the γ-Conservative Magician, our knapsack procedure selects for each query the sample paths with the *most* capacity utilized, on which that query still fits. We dub this procedure a "Best-fit Magician"[3]. In the more general knapsack setting, capacity utilization can only be tracked in polynomial time after discretizing sizes by $1/K$ for some large integer K; nonetheless we will show (in the full version [19]) that this loses a negligible additive term of $O(1/K)$ in the guarantee.

To analyze the maximum feasible guarantee γ for a Best-fit Magician, note that the expected capacity utilization over the sample paths is $\gamma \cdot \sum_t p_t d_t$, which is always upper-bounded by γ, since $\sum_t p_t d_t \le 1$. Therefore, to lower-bound the measure of sample paths with 0 utilization, it suffices to upper-bound the measure of sample paths whose utilization is small but non-zero. To do so, we use the rule of the Best-fit Magician—an arriving query with size d_t will only get packed on a previously-empty sample path if there is less than γ-measure of sample paths with utilization in $(0, 1 - d_t]$. Based on this fact, we derive an *invariant* which holds after each query t, and upper-bounds the measure of sample paths with utilization in $(0, b]$ by a decreasing exponential function of the measure with utilization in $(b, 1 - b]$, for any small size $b \in (0, 1/2]$. This allows us to show that a γ as large as $\frac{1}{3+e^{-2}} \approx 0.319$ allows for a γ-measure of sample paths to have 0 utilization at all times, and hence is feasible. The Best-fit Magician is also agnostic to knowing $\boldsymbol{p} = (p_1, \ldots, p_T)$ and $\boldsymbol{D} = (d_1, \ldots, d_T)$ in advance, as long as it is promised that $\sum_t p_t d_t \le 1$. Nonetheless, we construct a counterexample showing it to be *optimal*, in that

[3]This is because it resembles the "best-fit" heuristic for bin packing [16].

$\gamma = \frac{1}{3+e^{-2}}$ is an upper bound on the guarantee for the knapsack OCRS problem even if $\boldsymbol{p}$ and $\boldsymbol{D}$ are known in advance.

To our knowledge, our analysis differs from all existing ones for knapsack in an online setting [13, 15, 26] by eschewing the need to split queries into "large" vs. "small" based on its size (usually, whether its size is greater than 1/2). In fact, we show that any algorithm which *packs large and small queries separately* is limited to $\gamma \leq 0.25$ in our problem (Proposition 3.1), whereas our tight guarantee is $\gamma = \frac{1}{3+e^{-2}} \approx 0.319$.

Our result can be generalized to the setting where the size of a query is random and arbitrarily correlated with its reward . Furthermore, in the case of *unit-density* online knapsack, where the random size and reward of a query are always identical, our analysis can be improved to obtain an improved guarantee 0.3557. Finally, our analysis can be directly extended to the multi-resource setting where there are multiple resources, and each query can be assigned to be served by one resource. We refer the details of these extensions to the full version [19].

Comparison to [4]. Another related setting is the online stochastic generalized assignment problem of [4], for which the authors establish a guarantee of $1 - \frac{1}{\sqrt{k}}$, when each query can realize a random size that is at most $1/k$. They eliminate the possibility of "large" queries by imposing k to be at least 2, showing that a constant-factor guarantee is impossible when $k = 1$. Although our problem can be generalized to random sizes, we need to assume that size is observed *before* the algorithm makes a decision, whereas in their problem size is randomly realized *after* the algorithm decides to serve a query. This distinction allows our problem to have a constant-factor guarantee that holds even when queries can have size 1. Moreover, our procedure starkly contrasts theirs—we prioritize selecting sample paths with the *most* capacity utilized on which a query fits, while they prioritize sample paths with the *least* capacity utilized.

1.3 Roadmap Section 3 formalizes the model/notation for online knapsack with deterministic sizes, and presents our results in this setting. Section 4 presents all of our results for k-unit prophet inequalities, in which the sizes are always $1/k$.

2 Further Related Work

Online knapsack. The discrete version of the online stochastic knapsack problem as studied in this paper, which generalizes the k-unit prophet inequality problem, was originally posed in [25] to model classical applications in operations research such as freight transportation and scheduling. We derive the best competitive ratios for this problem known to date, for both the general and unit-density settings, which extend to multiple knapsacks. We should point out though that in the unit-density setting with a *single* knapsack, a competitive ratio of 1/2, better than our guarantee of 0.3557, is possible under any fixed sequence of adversarial arrivals [20]. However, such a guarantee fails[4] to extend to multiple knapsacks, whereas our guarantee of 0.3557 which holds relative to the LP directly extends, following the same reduction argument as in [26] (see the full version [19] for details).

Prophet inequalities. Prophet inequalities were originally posed in the statistics literature by [21]. Due to their implications for posted pricing and mechanism design, prophet inequalities have been a surging topic in algorithmic game theory since the seminal works of [2, 9, 18, 28]. Of particular interest in these works are bounds for k-item prophet inequalities, and in this paper we improve such bounds for all $k > 1$, and show that our bounds are tight relative to the ex-ante relaxation. In line with these works, our paper also focuses on the classical adversarial arrival order. More recently, prophet inequalities have also been studied under random order, free order, or IID arrivals, with k-unit prophet inequalities in particular being studied by [7] under free order. A survey of recent results in prophet inequalities can be found in [11].

Online Contention Resolution Schemes (OCRS). A guarantee of γ for our problem in Definition 1.1 (resp. Definition 1.2) is identical to a γ-*selectable OCRS* for the k-uniform matroid (resp. knapsack polytope) as introduced in [15]. However, we should clarify some assumptions about what is known beforehand and the choice of arrival order. Our OCRS's hold against an *online adversary*, who can adaptively choose the next query to arrive, but does not know the realizations of queries yet to arrive. We show that the guarantee does not improve against the *weakest adversary*, who has to reveal the arrival order in advance. However, our OCRS's do not satisfy the *greedy* property, and consequently do not hold against the *almighty adversary*, who knows the realizations of all queries before having to choose the order.

[4] An additional factor of 1/2 would be lost, resulting in a guarantee of only 1/4—see [23]. In fact, a guarantee of 1/2 relative to the LP is impossible, due to the upper bound of 0.432 presented in the full version [19]

In [15], the authors derive a 1/4-selectable[5] greedy OCRS for general matroids, and a 0.085-selectable greedy OCRS for the knapsack polytope, both of which hold against an almighty adversary. We establish significantly improved selectabilities against the weaker online adversary, and importantly, show that our guarantees are *tight* for our setting.

Magician's problem. Our algorithms do enjoy a property not featured in the OCRS setting though—they *need not know the universe* of elements in advance, holding even if the adversary can adaptively "create" the p_t (and d_t) of the next query t, under the promise that $\sum_t p_t d_t \leq 1$. This property is inherited from the *Magician's problem*, introduced by [2] as a powerful black box for approximately solving combinatorial auctions. Our work fully resolves[6] his k-unit Magician problem, showing his γ-Conservative Magician to be optimal, and importantly, showing how to find the optimal value $\gamma = \gamma_k^*$ which is greater than the value of $\gamma = 1 - \frac{1}{\sqrt{k+3}}$ for all $k > 1$. This improves all of the guarantees for combinatorial auctions, summarized in [5], which depend on this value of γ.

3 Online Knapsack with Deterministic Sizes

We consider an online stochastic single knapsack problem over T discrete time periods, where the initial knapsack capacity is without loss of generality scaled to be 1. At each time period t, a query arrives, denoted as query t, and is associated with a non-negative stochastic reward $\tilde{r}_t$ and a positive deterministic size d_t. We assume that the stochastic reward $\tilde{r}_t$ follows a known distribution $F_t(\cdot)$, which can be time in-homogeneous and is independent across time. After the value of $\tilde{r}_t$ is revealed, an online decision maker has to decide irrevocably whether to serve or reject this query. If query t is served, then the reward $\tilde{r}_t$ is collected and query t will take up d_t capacity of the knapsack. If rejected, then no reward is collected and no space in the knapsack will be consumed. The goal of the decision maker is to maximize the total collected reward subject to the capacity constraint of the knapsack.

We classify the elements described above into two categories:

(i). The problem setup $\mathcal{H} = (\boldsymbol{F}, \boldsymbol{D})$, where $\boldsymbol{F} = (F_1, F_2, \ldots, F_T)$ denotes all the distributions and $\boldsymbol{D} = (d_1, d_2, \ldots, d_T)$ denotes all the deterministic sizes.

(ii). The realization of the rewards of the arriving queries $\boldsymbol{I} = (\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_T)$.

Any online policy π for the decision maker can be specified by a set of decision variables $\{x_t^\pi\}_{t=1,\ldots,T}$, where x_t^π is a binary variable and denotes whether query t is served. Note that if π is a randomized policy, then x_t^π is random binary variable. A policy π is *feasible* if and only if π is non-anticipative, i.e., for each t, the distribution of x_t^π can only depend on the problem setup $\mathcal{H}$ and the rewards of the arriving queries up to query t, denoted by $\{\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_t\}$. Moreover, π needs to satisfy the following capacity constraint:

$$\sum_{t=1}^{T} d_t \cdot x_t^\pi \leq 1 \tag{3.1}$$

The total collected reward of policy π is denoted as $V^\pi(\boldsymbol{I}) = \sum_{t=1}^T \tilde{r}_t \cdot x_t^\pi$, and the expected total collected reward of policy π is denoted as $\mathbb{E}_{\pi, \boldsymbol{I} \sim \boldsymbol{F}}[V^\pi(\boldsymbol{I})]$.

We compare to the prophet, who can make decisions based on the knowledge of the realizations of all the queries. Similarly, the offline optimal decision is specified by $\{x_t^{\text{off}}(\boldsymbol{I})\}_{t=1,\ldots,T}$, which is the optimal solution to the following offline problem:

$$\begin{aligned} V^{\text{off}}(\boldsymbol{I}) = \ \max \quad & \sum_{t=1}^{T} \tilde{r}_t \cdot x_t \\ \text{s.t.} \quad & \sum_{t=1}^{T} d_t \cdot x_t \leq 1 \\ & x_t \in \{0, 1\}, \quad \forall t \end{aligned} \tag{3.2}$$

[5][22] have improved this to a 1/2-selectable OCRS for general matroids, against the weakest adversary. Our guarantees of γ_k^* are all greater than 1/2 and hold against the online adversary, but in the special case of k-uniform matroids.

[6]The main difference in the Magician's problem is that a query must be selected *before* knowing whether it is active, and if so, irrevocably served. The goal is to select each query with ex-ante probability at least γ. Our problem can be reinterpreted as selecting a γ-measure of sample paths on which each query should be served whenever it is active, which is completely equivalent. Therefore, all of our results also hold for Alaei's Magician problem and its applications.

The prophet's value is denoted as $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]$. For any feasible online policy π, its competitive ratio γ is defined as

$$\gamma = \inf_{\mathcal{H}} \frac{\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^{\pi}(\boldsymbol{I})]}{\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]} \tag{3.3}$$

In this paper, however, instead of directly comparing to $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]$, we compare to a linear programming (LP) upper bound of $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]$, which will provide more tractable structures of the optimal solution. We consider the following LP as an upper bound:

$$\begin{aligned} \mathrm{UP}(\mathcal{H}) = \ \max \ & \sum_{t=1}^{T} \mathbb{E}_{r_t\sim F_t}[r_t \cdot x_t(r_t)] \\ \text{s.t.} \ & \sum_{t=1}^{T} \mathbb{E}_{r_t\sim F_t}[d_t \cdot x_t(r_t)] \leq 1 \\ & 0 \leq x_t(r_t) \leq 1, \quad \forall t, \forall r_t \end{aligned} \tag{3.4}$$

which is defined on the problem setup $\mathcal{H}$. Here, the variable $x_t(r_t)$ denotes the probability of serving query t conditional on its reward realizing to r_t. Due to the following lemma, in order to show a policy π has a competitive ratio at least γ, it suffices to show that $\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^{\pi}(\boldsymbol{I})]$ is at least γ times the optimal fractional allocation value $\mathrm{UP}(\mathcal{H})$ for every setup $\mathcal{H}$.

Lemma 3.1. *For all problem setups $\mathcal{H}$, it holds that $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})] \leq UP(\mathcal{H})$.*

The proof of Lemma 3.1 is relegated to the Section A. In what follows, we will derive our policy based on the optimal solution of $\mathrm{UP}(\mathcal{H})$. We first without losing generality reduce the distribution $F_t(\cdot)$ to a two-point distribution for each t, then we motivate our algorithm from the dynamic programming (DP) of the simplified problem. We obtain a LP formulation of the DP and show its special structure to motivate our algorithm. This structure is further exploited under a special case where $d_t = 1/k$ for each t in Section 4.

3.1 Reduction to two-point distributions, LP duality, and upper bound. In this section we first show that we can equivalently reduce $F_t(\cdot)$ for each t to a two-point distribution. Assuming two-point distributions, we then derive our new LP formulation of the knapsack OCRS problem through duality. Finally, using this duality we establish an upper bound on the performance relative to the fractional relaxation in the knapsack problem.

For any problem setup $\mathcal{H}$, denote $\{x_t^*(r_t)\}_{t=1,\dots,T}$ as one optimal solution of $\mathrm{UP}(\mathcal{H})$. Fixing this optimal solution, for each t, define

$$p_t = \mathbb{E}_{r_t\sim F_t}[x_t^*(r_t)] \ \text{ and } \ \hat{r}_t = \frac{\mathbb{E}_{r_t\sim F_t}[r_t \cdot x_t^*(r_t)]}{p_t} \tag{3.5}$$

and define a two-point distribution $\hat{F}_t(\cdot)$ with support $\{0, \hat{r}_t\}$, where $\hat{F}_t(0) = 1 - p_t$ (note that $\hat{r}_t$ does not need to be defined if $p_t = 0$). With these notations, we reduce to a problem setup $\hat{\mathcal{H}} = (\hat{\boldsymbol{F}}, \boldsymbol{D})$, where $\hat{\boldsymbol{F}} = (\hat{F}_1, \dots, \hat{F}_T)$. It is direct to check that $\sum_{t=1}^{T} p_t \cdot d_t \leq 1$, thus we have $\mathrm{UP}(\hat{\mathcal{H}}) = \sum_{t=1}^{T} \hat{r}_t \cdot p_t$. The next lemma shows that such reduction will not influence the ratio we can obtain, thus, it is enough to only focus on the case where the distributions of the reward are two-point distributions.

Lemma 3.2. *An online policy π satisfying $\mathbb{E}_{\pi,\hat{\boldsymbol{I}}\sim\hat{\boldsymbol{F}}}[V^{\pi}(\hat{\boldsymbol{I}})] \geq \gamma \cdot UP(\hat{\mathcal{H}})$ for all two-point distribution case $\hat{\mathcal{H}}$ can be translated into an online policy π' with a competitive ratio at least γ.*

We should note that the sufficiency of two-point distributions for establishing ex-ante prophet inequalities has already been observed in [22]. Nonetheless, for completeness we provide a self-contained proof of Lemma 3.2 in Section B.

In what follows, we focus on the two-point distribution case $\hat{\mathcal{H}}$. We first upper-bound the guarantee of any online algorithm by considering the dynamic programming formulation of the optimal online policy. Denote

$V_t^*(c_t)$ as the "value-to-go" function at period t where the remaining capacity is c_t, we have the following backward induction

$$V_t^*(c_t) = V_{t+1}^*(c_t) + p_t \cdot \underbrace{\max\left\{0, 1_{\{c_t \geq d_t\}} \cdot (\hat{r}_t + V_{t+1}^*(c_t - d_t) - V_{t+1}^*(c_t))\right\}}_{\text{marginal increase}} \quad \text{for all } t,\ 0 \leq c_t \leq 1 \tag{3.6}$$

where $V_{T+1}^*(\cdot) = 0$. Having this notation, we are interested in characterizing the quantity

$$\inf_{\hat{\mathcal{H}}} \frac{V_1^*(1)}{\mathrm{UP}(\hat{\mathcal{H}})}. \tag{3.7}$$

Note that $\hat{\mathcal{H}}$, as defined by the values of $\hat{r}_t$, p_t, and d_t, must necessarily satisfy $\sum_{t=1}^T p_t \cdot d_t \leq 1$. Moreover, it is without loss of generality to scale $\hat{\boldsymbol{r}} = (\hat{r}_1, \ldots, \hat{r}_T)$ such that $\mathrm{UP}(\hat{\mathcal{H}}) = \sum_{t=1}^T p_t \cdot \hat{r}_t = 1$. Therefore, we can re-express (3.7) as the infimum value of the following LP over all vectors $\boldsymbol{p} = (p_1, \ldots, p_T), \boldsymbol{D} = (d_1, \ldots, d_T)$ satisfying $\sum_{t=1}^T p_t \cdot d_t \leq 1$.

$$\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D}) = \min \quad V_1(1) \tag{3.8}$$
$$\text{s.t.} \quad V_t(c_t) \geq V_{t+1}(c_t) + p_t \cdot W_t(c_t), \quad \forall t, \forall c_t \tag{3.8a}$$
$$W_t(c_t) \geq \hat{r}_t + V_{t+1}(c_t - d_t) - V_{t+1}(c_t), \quad \forall t, \forall c_t \geq d_t \tag{3.8b}$$
$$\sum_{t=1}^T p_t \cdot \hat{r}_t = 1 \tag{3.8c}$$
$$V_{T+1}(c) = 0, \quad \forall c$$
$$V_t(c_t), W_t(c_t) \geq 0, \hat{r}_t \geq 0 \quad \forall t, \forall c_t$$

In this LP $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$, decision variable $V_t(c_t)$ represents the dynamic programming value-to-go $V_t^*(c_t)$, decision variable $W_t(c_t)$ represents the marginal increase term, and decision variable $\hat{r}_t$ must be scaled to respect $\sum_{t=1}^T p_t \cdot \hat{r}_t = 1$. It is known that in an optimal solution, the decision variables $V_t(c_t)$ equal $V_t^*(c_t)$ [1]. With some further massaging, we obtain an LP dual with the interpretable set of constraints (3.9b) below.

$$\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D}) = \max \quad \theta \tag{3.9}$$
$$\text{s.t.} \quad \theta \cdot p_t \leq \sum_{c_t \geq d_t} \alpha_t(c_t), \quad \forall t \tag{3.9a}$$
$$\alpha_t(c_t) \leq p_t \cdot \sum_{\tau \leq t-1} (\alpha_\tau(c_t + d_\tau) - \alpha_\tau(c_t)), \quad \forall t, \forall c_t < 1 \tag{3.9b}$$
$$\alpha_t(1) \leq p_t \cdot (1 - \sum_{\tau \leq t-1} \alpha_\tau(1)), \quad \forall t \tag{3.9c}$$
$$\alpha_t(c_t) = 0, \quad \forall t, \forall c_t > 1$$
$$\theta, \alpha_t(c_t) \geq 0, \quad \forall t, \forall c_t$$

In fact, $\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$ characterizes the knapsack OCRS problem from the Introduction, where each query t has a size d_t and becomes active with probability p_t. To see this, we interpret the variable θ as the guarantee γ in the knapsack OCRS problem, and we interpret the variable $\alpha_t(c_t)$ for each t and each $0 \leq c_t \leq 1$ as the *ex-ante* probability (the unconditional probability) that query t is served when the remaining capacity is c_t at the beginning of period t. Then, constraint (3.9a) corresponds to each query t being served with probability at least $\theta = \gamma$ conditional on it being active. Moreover, for each t and each $0 \leq c_t < 1$, the term $\sum_{\tau=1}^{t-1} \alpha_\tau(c_t + d_\tau)$ denotes the ex-ante probability that the remaining capacity has "reached" the state c_t in the first $t-1$ periods and the term $\sum_{\tau=1}^{t-1} \alpha_\tau(c_t)$ denotes the ex-ante probability that the remaining capacity has "left" the state c_t. Thus, the difference $\sum_{\tau=1}^{t-1}(\alpha_\tau(c_t + d_\tau) - \alpha_\tau(c_t))$ denotes exactly the ex-ante probability that the remaining capacity at the beginning of period t is c_t. Similarly, the difference $1 - \sum_{\tau=1}^{t-1} \alpha_\tau(1)$ denotes the ex-ante probability that the remaining capacity at the beginning of period t is 1. Since each query t can only be served after being active, which happens independently with probability p_t, we recover constraint (3.9b) and constraint (3.9c).

Consequently, the worst-case guarantee for the knapsack OCRS problem is equivalent to the worst-case guarantee relative to the fractional relaxation in the online knapsack problem. The following lemma formalizes this fact. Its proof is relegated to Section C, but following the discussion above, the only part that requires proof is establishing the duality between the LP's Primal$(\boldsymbol{p}, \boldsymbol{D})$ and Dual$(\boldsymbol{p}, \boldsymbol{D})$.

Lemma 3.3. *It holds that*

$$\inf_{\hat{\mathcal{H}}} \frac{V_1^*(1)}{UP(\hat{\mathcal{H}})} = \inf_{(\boldsymbol{p},\boldsymbol{D}):\sum_{t=1}^T p_t \cdot d_t \leq 1} Primal(\boldsymbol{p}, \boldsymbol{D}) = \inf_{(\boldsymbol{p},\boldsymbol{D}):\sum_{t=1}^T p_t \cdot d_t \leq 1} Dual(\boldsymbol{p}, \boldsymbol{D}).$$

Using Lemma 3.3, we can derive upper bounds on the performance of any online policy relative to UP$(\hat{\mathcal{H}})$. Specifically, by bounding the optimal value of Dual$(\boldsymbol{p}, \boldsymbol{D})$ when $\boldsymbol{p}$ and $\boldsymbol{D}$ are specified as follows,

$$(p_1, d_1) = (1, \epsilon), \quad (p_t, d_t) = (\frac{1-2\epsilon}{(T-2)(\frac{1}{2}+\epsilon)}, \frac{1}{2}+\epsilon) \text{ for all } 2 \leq t \leq T-1 \text{ and } (p_T, d_T) = (\epsilon, 1) \tag{3.10}$$

for some $\epsilon > 0$, we obtain the following upper bound of the guarantee of any online algorithm.

Theorem 3.1. *For any feasible online policy π, it holds that* $\inf_{\hat{\mathcal{H}}} \frac{\mathbb{E}_{\hat{\boldsymbol{I}}\sim\hat{\boldsymbol{F}}}[V^\pi(\hat{\boldsymbol{I}})]}{UP(\hat{\mathcal{H}})} \leq \frac{1}{3+e^{-2}}$.

Proof. Due to Lemma 3.3, it is enough to construct a $\boldsymbol{p}$ and $\boldsymbol{D}$ satisfying $\sum_{t=1}^T p_t \cdot d_t \leq 1$, such that Dual$(\boldsymbol{p}, \boldsymbol{D}) \leq \frac{1}{3+e^{-2}}$. For each $\epsilon > 0$, we consider the following $\boldsymbol{p}$ and $\boldsymbol{D}$:

$$(p_1, d_1) = (1, \epsilon), \quad (p_t, d_t) = (\frac{1-2\epsilon}{(T-2)(\frac{1}{2}+\epsilon)}, \frac{1}{2}+\epsilon) \text{ for all } 2 \leq t \leq T-1 \text{ and } (p_T, d_T) = (\epsilon, 1)$$

It is direct to check that $\sum_{t=1}^T p_t \cdot d_t = 1$. Now denote $\{\theta^*, \alpha_t^*(c_t)\}$ as the optimal solution to Dual$(\boldsymbol{p}, \boldsymbol{D})$. We further denote $\mu_t = \sum_{c_t: d_t \leq c_t < 1} \alpha_t^*(c_t)$ for each $2 \leq t \leq T-1$. Then constraint (3.9a) implies that

$$\alpha_t^*(1) \geq \theta^* \cdot p_t - \mu_t, \quad \forall 2 \leq t \leq T-1$$

Thus, we have that

$$\theta^* \cdot p_T \leq \alpha_T^*(1) \leq p_T \cdot (1 - \alpha_1^*(1) - \sum_{t=2}^{T-1} \alpha_t^*(1)) \Rightarrow \theta^* \cdot (1 + \sum_{t=2}^{T-1} p_t) \leq 1 - \alpha_1^*(1) + \sum_{t=2}^{T-1} \mu_t$$

We then consider the term $\sum_{t=2}^{T-1} \mu_t$ to upper bound θ^*. Note that since $d_t > 1/2$ for all $2 \leq t \leq T-1$, constraint (3.9b) implies that

$$\alpha_t^*(c_t) \leq p_t \cdot (\alpha_1^*(c_t + d_1) - \sum_{\tau=2}^{t-1} \alpha_\tau^*(c_t)), \quad \forall d_t \leq c_t < 1$$

Further note that $\alpha_1^*(c) = 0$ when $c < 1$, we must have that $\alpha_t^*(c_t) = 0$ when $c_t \neq 1 - d_1$. Then, we have $\mu_t = \alpha_t^*(1 - d_1)$ and

$$\mu_t \leq p_t \cdot (\alpha_1^*(1) - \sum_{\tau=2}^{t-1} \mu_\tau), \quad \forall 2 \leq t \leq T-1$$

We then inductively show that for any $2 \leq t \leq T-1$, we have

$$\sum_{\tau=2}^{t} \mu_t \leq (1 - \prod_{\tau=2}^{t} (1 - p_\tau)) \cdot \alpha_1^*(1) \tag{3.11}$$

When $t = 2$, (3.11) holds obviously. Now suppose that (3.11) holds for $t-1$, we have

$$\sum_{\tau=2}^{t} \mu_t = \sum_{\tau=2}^{t-1} \mu_\tau + \mu_t \leq \alpha_1^*(1) \cdot p_t + (1-p_t) \cdot \sum_{\tau=2}^{t-1} \mu_\tau \leq \alpha_1^*(1) \cdot p_t + (1-p_t) \cdot (1 - \prod_{\tau=2}^{t-1} (1-p_\tau)) \cdot \alpha_1^*(1)$$
$$= \alpha_1^*(1) \cdot (1 - \prod_{\tau=2}^{t} (1-p_\tau))$$

Thus, (3.11) holds for all $2 \le t \le T-1$ and we have

$$\sum_{t=2}^{T-1} \mu_t \le \alpha_1^*(1) \cdot (1 - \prod_{t=2}^{T-1}(1-p_t)) = \alpha_1^*(1) \cdot (1-(1-\frac{1-2\epsilon}{(T-2)(\frac{1}{2}+\epsilon)})^{T-2}) \le \alpha_1^*(1)\cdot(1-\exp(-2)) + O(\epsilon)$$

which implies that

$$3\theta^* + O(\epsilon) = \theta^* \cdot (1 + \sum_{t=2}^{T-1} p_t) \le 1 - \alpha_1^*(1) + \sum_{t=2}^{T-1} \mu_t \le 1 - \alpha_1^*(1) \cdot \exp(-2) + O(\epsilon)$$

Moreover, note that constraint (3.9a) requires $\alpha_1^*(1) \ge \theta^*$, we have that $\theta^* \le \frac{1}{3+e^{-2}} + O(\epsilon)$. The proof is finished by taking $\epsilon \to 0$. $\square$

3.2 Algorithm. In this section, we derive our algorithm with a tight guarantee. Our algorithm differs from existing ones for knapsack in an online setting [13, 15, 26] by eschewing the need to split queries into "large" vs. "small" based on whether its size is greater than 1/2. In fact, we first show that any algorithm which *considers large and small queries separately* in our problem is limited to $\gamma \le 1/4$, and hence could not match the $\frac{1}{3+e^{-2}}$ upper bound provided earlier. The result follows by considering a problem setup $\hat{\mathcal{H}}$ with 4 queries and

$$(\hat{r}_1, p_1, d_1) = (r, 1, \epsilon), \quad (\hat{r}_2, p_2, d_2) = (\hat{r}_3, p_3, d_3) = (r, \frac{1-2\epsilon}{1+2\epsilon}, \frac{1}{2}+\epsilon), \quad (\hat{r}_4, p_4, d_4) = (r/\epsilon, \epsilon, 1)$$

for $r > 0$ and some small $\epsilon > 0$. The formal proof is contained in Section D.

PROPOSITION 3.1. *If the policy π serves only either "large" queries with a size larger than 1/2, or "small" queries with a size no larger than 1/2, then it holds that* $\inf_{\hat{\mathcal{H}}} \frac{\mathbb{E}_{\hat{\boldsymbol{I}} \sim \hat{\boldsymbol{F}}}[V^\pi(\hat{\boldsymbol{I}})]}{UP(\hat{\mathcal{H}})} \le \frac{1}{4}$.

We now motivate our algorithm. For any problem setup $\hat{\mathcal{H}}$, if we call a query "active" when the reward of this query is realized to be non-zero, then we recover the knapsack OCRS problem from the Introduction; we will proceed with that terminology. Our approach for the knapsack OCRS problem is based on tracking the distribution of capacity utilization over time, and for each query t, selecting a γ-measure of sample paths on which it should be served whenever active. This corresponds to constraint (3.9a) in Dual$(\boldsymbol{p}, \boldsymbol{D})$ defined in (3.9), where the variable $\alpha_t(c_t)$ denotes the measure of sample paths selected with a remaining capacity c_t at the beginning of period t. Specifically, our approach selects for each query the sample paths with the *most* capacity utilized, *on which that query still fits.* Our approach also admits an interpretation as packing rectangles into a 1×1 block as we will later demonstrate through an example.

Now we formally describe our policy for a general $\boldsymbol{p}$ and $\boldsymbol{D}$ in Algorithm 1. To be specific, we use $\tilde{X}_{t-1,\gamma}$ to denote the distribution of the capacity consumption under our policy π_γ at the beginning of period t, where $\tilde{X}_{0,\gamma}$ takes value 0 deterministically. Note that since each query has a deterministic size, the random variable $\tilde{X}_{t-1,\gamma}$ must have a finite support and our policy iteratively keeps track of the distribution of $\tilde{X}_{t-1,\gamma}$ for each t. Then, we specify a threshold $\eta_{t,\gamma}$ such that the probability of $\tilde{X}_{t-1,\gamma} \in (\eta_{t,\gamma}, 1-d_t]$ is smaller than or equal to γ, and the probability that $\tilde{X}_{t-1,\gamma} \in [\eta_{t,\gamma}, 1-d_t]$ is larger than or equal to γ. When query t becomes "active", we only serve query t when the realized capacity consumption is among $(\eta_{t,\gamma}, 1-d_t]$, or we serve query t with a certain probability (specified in step 5) when the realized capacity consumption equals $\eta_{t,\gamma}$, to guarantee that query t is served with a total probability γ conditional on being "active". We finally update the distribution of capacity consumption in step 6. Note that here, we establish competitive ratios ignoring implementation runtime of step 6; in the full version [19], we show how through discretization, a runtime polynomial in K is attainable while losing only an additive $O(1/K)$ in the competitive ratio, for any large integer K.

We illustrate the implementation of our policy through the following example:

Example 1. Consider a problem setup where $T = 4$. We set

$$(p_1, d_1) = (\frac{2}{3}, \frac{1}{2}), \ (p_2, d_2) = (\frac{2}{3}, \frac{1}{2}), \ (p_3, d_3) = (1-\epsilon, \frac{1}{3}) \text{ and } (p_4, d_4) = (\frac{\epsilon}{3}, 1)$$

Algorithm 1 Best-fit Magician Policy (π_γ)

1: For a fixed γ, initialize $\tilde{X}_{0,\gamma}$ as a random variable which takes the value 0 deterministically.
2: **For** $t = 1, 2, \ldots, T$, do the following:
3: denote $\{b_1, b_2, \ldots, b_M\}$ as the support of $\tilde{X}_{t-1,\gamma}$ in an increasing order and denote $v_j = P(\tilde{X}_{t-1,\gamma} = b_j)$ for each $j = 1, \ldots, M$.
4: denote t_2 as the largest index such that $b_{t_2} + d_t \le 1$ and define threshold $\eta_{t,\gamma}$ such that

$$P(\eta_{t,\gamma} < \tilde{X}_{t-1,\gamma} \le 1 - d_t) \le \gamma \le P(\eta_{t,\gamma} \le \tilde{X}_{t-1,\gamma} \le 1 - d_t) \tag{3.12}$$

denote t_1 as the index of $\eta_{t,\gamma}$ in the support set $\{b_1, b_2, \ldots, b_M\}$ such that $\eta_{t,\gamma} = b_{t_1}$.
5: denote X_{t-1} as the consumed space of the knapsack at the end of period $t-1$. When $b_{t_1+1} \le X_{t-1} \le b_{t_2}$, then serve query t as long as it becomes active. When $X_{t-1} = b_{t_1}$, then serve query t with probability $(\gamma - \sum_{j=t_1+1}^{t_2} v_j)/v_{t_1}$ when it becomes active.
6: for each $t_1 + 1 \le j \le t_2$, move $v_j \cdot p_t$ probability mass of $\tilde{X}_{t-1,\gamma}$ at point b_j to the point $b_j + d_t$, and move $p_t \cdot (\gamma - \sum_{j=t_1+1}^{t_2} v_j)$ probability mass of $\tilde{X}_{t-1,\gamma}$ at point b_{t_1} to the point $b_{t_1} + d_t$ to obtain a random variable $\tilde{X}_{t,\gamma}$. Specifically, we first initialize the random variable $\tilde{X}_{t,\gamma}$ with the same distribution as $\tilde{X}_{t-1,\gamma}$, then, for each $j = t_1, t_1+1, \ldots, t_2$, we sequentially do the following update:

$$\begin{aligned}
&P(\tilde{X}_{t,\gamma} = b_{t_1}) = P(\tilde{X}_{t,\gamma} = b_{t_1}) - p_t \cdot (\gamma - \sum_{j=t_1+1}^{t_2} v_j), && \text{if } j = t_1 \\
&P(\tilde{X}_{t,\gamma} = b_j) = P(\tilde{X}_{t,\gamma} = b_j) - v_j \cdot p_t, && \text{if } t_1 + 1 \le j \le t_2 \\
&P(\tilde{X}_{t,\gamma} = b_j + d_t) = P(\tilde{X}_{t,\gamma} = b_j + d_t) + v_j \cdot p_t, && \text{if } t_1 + 1 \le j \le t_2
\end{aligned} \tag{3.13}$$

where $\epsilon > 0$. After the ratio γ is fixed, our approach serves each query t with an ex-ante probability $\gamma \cdot p_t$ whenever this query is active. The random variable $\tilde{X}_{t,\gamma}$ is introduced to denote the distribution of capacity utilization at the end of period t. Specifically, we set $\tilde{X}_{0,\gamma} = 0$.

The first query is accepted with probability γ whenever it is active. Then $\tilde{X}_{1,\gamma}$ equals $d_1 = \frac{2}{3}$ with probability $\gamma \cdot p_1$ and equals 0 with probability $1 - \gamma \cdot p_1$. For the second query, we first consider serving it on the sample paths of $\tilde{X}_{1,\gamma}$ with the largest capacity utilization. To be more specific, conditioning on $\tilde{X}_{1,\gamma} = d_1$, query 2 is served whenever it is active. Obviously, since the distribution of $\tilde{X}_{2,\gamma}$ is independent of whether query 2 is active, we guarantee that query 2 is served with an ex-ante probability $p_2 \cdot P(\tilde{X}_{1,\gamma} = d_1) = \gamma \cdot p_1 \cdot p_2$. We then serve query 2 on the remaining sample paths of $\tilde{X}_{1,\gamma}$ with 0 capacity utilization such that in total, query 2 is served with an ex-ante probability $\gamma \cdot p_2$. Accordingly, we move $\gamma \cdot p_1 \cdot p_2$-measure of sample paths of $\tilde{X}_{1,\gamma}$ from d_1 to $d_1 + d_2$ and move $\gamma \cdot (1 - p_1) \cdot p_2$-measure of sample paths of $\tilde{X}_{1,\gamma}$ from 0 to d_2, then we get $\tilde{X}_{2,\gamma}$, denoting the distribution of capacity utilization at the end of period 2. For the third query, since it is not feasible to serve it when $\tilde{X}_{2,\gamma} = d_1 + d_2$, we serve it whenever $\tilde{X}_{2,\gamma} = d_1 = d_2$ and query 3 is active. We then serve query 3 on the remaining sample paths of $\tilde{X}_{1,\gamma} = 0$ such that in total, query 3 is served with an ex-ante probability $\gamma \cdot p_3$. Accordingly, from $\tilde{X}_{2,\gamma}$, we obtain $\tilde{X}_{3,\gamma}$, denoting the distribution of capacity utilization at the end of period 3. For the last query, it is only feasible to serve it whenever $\tilde{X}_{3,\gamma} = 0$ and query 4 is active. Thus, we serve query 4 with an ex-ante probability at most $p_4 \cdot P(\tilde{X}_{3,\gamma} = 0)$. Note that our approach guarantees each query t is served with an ex-ante probability $\gamma \cdot p_t$. It must hold that

$$P(\tilde{X}_{3,\gamma} = 0) = 1 - \frac{8\gamma}{9} - \frac{5\gamma(1-\epsilon)}{9} \ge \gamma \tag{3.14}$$

which implies the feasibility condition on γ.

We can also interpret the above procedure by filling up a 1×1 block using four rectangles with height $\gamma \cdot p_t$ and length d_t for $t = 1, 2, 3, 4$, where each rectangle can be further split horizontally into multiple rectangles to fit the space better. Indeed, our method puts the first $\gamma \cdot p_1 \times d_1$ rectangle at the left-bottom corner and then the height of the non-empty row of the block corresponds to the probability that $\tilde{X}_{1,\gamma} = d_1$. For the second query, since $d_1 + d_2 \leq 1$, we put it to the right of the first rectangle. However, since query 2 becomes active with probability p_2, we can put only a height $\gamma \cdot p_1 \cdot p_2$ length d_2 rectangle to the right of first query, where this height corresponds to the probability that $\tilde{X}_{2,\gamma} = d_1 + d_2$, and we can only put the remaining height $\gamma \cdot (1 - p_1) \cdot p_2$ length d_2 rectangle to the top of the first query. The above procedure is repeated for the third query and the fourth query. See Figure 2 for a detailed description. Finally, in order to make the above procedure feasible, it is enough to guarantee that $P(\tilde{X}_{3,\gamma} = 0) \geq \gamma$, which implies that the height of the non-empty rows in the 1×1 block is no larger than $1 - \gamma$ at the end of period 3. This condition enables us to find the best-possible γ for our procedure. Moreover, this condition also explains why for each query we fit into the sample paths with a decreasing order of capacity utilization, we do so to keep the height of the non-empty rows in the 1×1 block as small as possible. $\square$

In fact, the policy π_γ constructs a feasible solution to Dual$(\boldsymbol{p}, \boldsymbol{D})$ in (3.9). Specifically, we denote variable θ in Dual$(\boldsymbol{p}, \boldsymbol{D})$ as γ in our policy π_γ. Then, for each t and each $0 \leq c_t \leq 1$, we denote the variable $\alpha_t(c_t)$ in Dual$(\boldsymbol{p}, \boldsymbol{D})$ as the ex-ante probability that query t is served when the *remaining capacity* is c_t at the beginning of period t, which also denotes the amount of probability mass of $\tilde{X}_{t-1,\gamma}$ that is moved from point $1 - c_t$ to the point $1 - c_t + d_t$ when defining $\tilde{X}_{t,\gamma}$ in (3.13). Obviously, when γ is feasible, the policy π_γ guarantees that each query t is served with an ex-ante probability $\gamma \cdot p_t$, which corresponds to constraint (3.9a). Moreover, from (3.13), it is clear to see that

$$P(\tilde{X}_{t,\gamma} = 0) = P(\tilde{X}_{t-1,\gamma} = 0) - \alpha_t(1)$$

and

$$P(\tilde{X}_{t,\gamma} = 1 - c) = P(\tilde{X}_{t-1,\gamma} = 1 - c) + \alpha_t(c + d_t) - \alpha_t(c), \quad \forall c < 1$$

which implies that

$$P(\tilde{X}_{t-1,\gamma} = 0) = 1 - \sum_{\tau=1}^{t-1} \alpha_\tau(1)$$

and

$$P(\tilde{X}_{t-1,\gamma} = 1 - c) = \sum_{\tau=1}^{t-1} (\alpha_\tau(c + d_\tau) - \alpha_\tau(c)), \quad \forall c < 1$$

Then, the constraint $\alpha_t(c) \leq p_t \cdot P(\tilde{X}_{t-1,\gamma} = 1 - c)$ for each $0 \leq c \leq 1$ in the policy π_γ corresponds to constraint (3.9b) and constraint (3.9c).

3.3 Proof of competitive ratio. In this section, we analyze the competitive ratio of our Best-fit Magician policy in Algorithm 1. Obviously, when γ is fixed, policy π_γ guarantees that the decision maker collects at least γ times the optimal fractional allocation UP$(\hat{\mathcal{H}})$. From Lemma 3.3, it is enough to find the largest possible γ such that the policy π_γ is feasible for all problem setups $\hat{\mathcal{H}}$, i.e., random variables $\tilde{X}_{t,\gamma}$ are well-defined for each t.

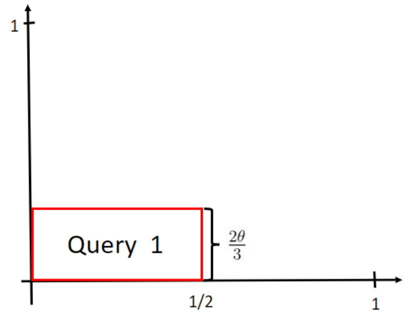
We now find such γ. For any a and b, denote $\mu_{t,\gamma}(a, b] = P(a < \tilde{X}_{t,\gamma} \leq b)$ assuming $\tilde{X}_{t,\gamma}$ is well-defined. A key observation of the Best-fit Magician is that an arriving query with size d_t will only get packed on a previously-empty sample path if there is less than γ-measure of sample paths with utilization in $(0, 1 - d_t]$. Then, we can establish an *invariant* that upper-bounds the measure of sample paths with utilization in $(0, b]$ by a decreasing exponential function of the measure with utilization in $(b, 1 - b]$. Our invariant holds for all $b \in (0, 1/2]$, at all times t.

LEMMA 3.4. *For any $0 < b \leq \frac{1}{2}$, as long as $\tilde{X}_{t,\gamma}$ is well-defined, the following inequality*

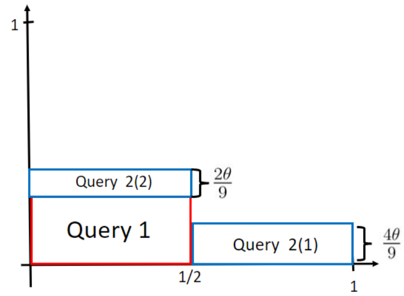
$$\frac{1}{\gamma} \cdot \mu_{t,\gamma}(0, b] \leq \exp(-\frac{1}{\gamma} \cdot \mu_{t,\gamma}(b, 1 - b]) \tag{3.15}$$

holds for all $t = 0, 1, \ldots, T$.

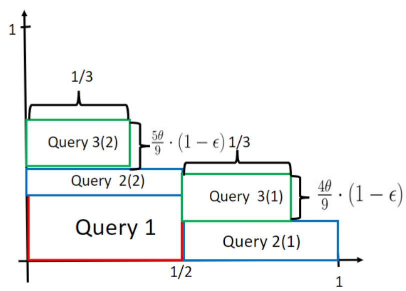
Proof. We prove (3.15) by induction on t. When $t = 0$, since $\mu_{0,\gamma}(0, b] = 0$ for any γ and any $0 < b \leq 1/2$, (3.15) holds trivially. Now suppose that (3.15) holds for $t - 1$, we consider the evolution of the distribution of $\tilde{X}_{t,\gamma}$ in (3.13) to obtain the relationship between $\mu_{t,\gamma}(0, b]$ and $\mu_{t,\gamma}(b, 1 - b])$ for each $b \in (0, 1/2]$.



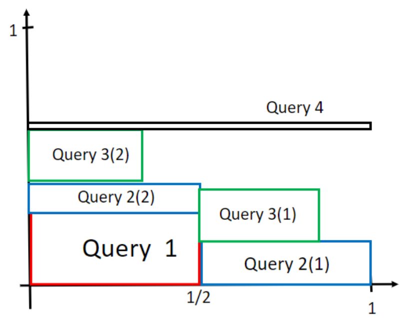
(a) When $t = 1$, our approach is equivalent to putting a $\frac{2\gamma}{3} \times \frac{1}{2}$ rectangle to the left-bottom corner of the 1×1 block, where the height $\frac{2\gamma}{3}$ corresponds to the measure of sample paths that query 1 is served and the length $\frac{1}{2}$ corresponds to the size d_1. $\tilde{X}_{1,\gamma}$ is a random variable denoting the distribution of capacity utilization, which also expresses the height of the rows with different occupied length. Here, $P(\tilde{X}_{1,\gamma} = \frac{1}{2}) = \frac{2\gamma}{3}$ and $P(\tilde{X}_{1,\gamma} = 0) = 1 - \frac{2\gamma}{3}$.



(b) When $t = 2$, we first fit to the right of the first query, however, the height of query 2(1) can only be p_2 fraction of the height of query 1, thus, query 2(1) is a $\frac{4\gamma}{9} \times \frac{1}{2}$ rectangle, where the height $\frac{4\gamma}{9} = \gamma \cdot p_1 \cdot p_2$. We put the remaining $\frac{2\gamma}{9} \times \frac{1}{2}$ to the top of query 1. Query 2(1) and query 2(2) combined will become a $\frac{2\gamma}{3} \times \frac{1}{2}$ rectangle, which corresponds to query 2. Here, $P(\tilde{X}_{2,\gamma} = 1) = \frac{4\gamma}{9}$, $P(\tilde{X}_{2,\gamma} = \frac{1}{2}) = \frac{4\gamma}{9}$ and $P(\tilde{X}_{2,\gamma} = 0) = 1 - \frac{8\gamma}{9}$



(c) When $t = 3$, there is no enough space to fit to the right of query 2(1), thus, we fit to the top of query 2(1), i.e., we fit to the right of the rows with occupied length 1/2. Note that the total height of the rows with occupied length 1/2 is $\frac{4\gamma}{9}$, the height of the fitted query 3(1) can be at most $\frac{4\gamma}{9} \cdot (1 - \epsilon)$. The remaining $\frac{5\gamma}{9} \cdot (1 - \epsilon) \times \frac{1}{3}$ rectangle is fitted to the top of query 2(2). Here, $P(\tilde{X}_{3,\gamma} = 1) = \frac{4\gamma}{9}$, $P(\tilde{X}_{3,\gamma} = \frac{5}{6}) = \frac{4\gamma(1-\epsilon)}{9}$, $P(\tilde{X}_{3,\gamma} = \frac{1}{2}) = \frac{4\gamma\epsilon}{9}$, $P(\tilde{X}_{3,\gamma} = \frac{1}{3}) = \frac{5\gamma(1-\epsilon)}{9}$ and $P(\tilde{X}_{3,\gamma} = 0) = 1 - \frac{8\gamma}{9} - \frac{5\gamma(1-\epsilon)}{9}$.



(d) When $t = 4$, we fit in the last query. Since $d_4 = 1$, we can only fit to the top of query 3(2) with a height $\gamma p_4 = \frac{\gamma\epsilon}{3}$. Note that in order for γ being feasible, we must have this height is no larger than p_4 fraction of the height of the rows with 0 occupied length, which implies that our approach is feasible as long as the height of the rows with 0 occupied length is at least γ, i.e., $P(\tilde{X}_{3,\gamma} = 0) \geq \gamma$.

Figure 2: **A graph illustration of the Best-fit Magician policy on Example** 1.

Note that if the threshold $\eta_{t,\gamma}$, defined in (3.12), is larger than 0, then no probability mass of $\tilde{X}_{t-1,\gamma}$ is moved from 0 to d_t. Thus, the probability mass over the interval $(0, b]$ can only become smaller in the update (3.13). On the contrary, if the threshold $\eta_{t,\gamma}$ equals 0, then a certain probability mass of $\tilde{X}_{t-1,\gamma}$ is moved from 0 to d_t, which can fall in either the interval $(0, b]$ or $(b, 1-b]$ when $d_t \leq 1-b$ to make the probability mass over the corresponding interval become larger. Thus, we consider the following separate cases depending on how the probability mass of $\tilde{X}_{t-1,\gamma}$ on 0 is moved in the update (3.13).

Case 1: When $\eta_{t.\gamma} > 0$, then no probability mass of $\tilde{X}_{t-1,\gamma}$ is moved from 0 to d_t. As a result, the probability mass over the interval $(0, b]$ can only become smaller. We denote

$$\mu_{t,\gamma}(0, b] = \mu_{t-1,\gamma}(0, b] - \tilde{p}_{t,\gamma}$$

for some $\tilde{p}_{t,\gamma} \geq 0$. Moreover, in terms of the interval $(b, 1-b]$, the probability mass that is moved into this interval in the update (3.13) can only come from the interval $(0, b]$, thus, we must have

$$\mu_{t,\gamma}(b, 1-b] \leq \mu_{t-1,\gamma}(b, 1-b] + \tilde{p}_{t,\gamma}$$

Then, we have that

$$\begin{aligned}
\exp(-\frac{1}{\gamma} \cdot \mu_{t,\gamma}(b, 1-b]) &\geq \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b] - \frac{1}{\gamma} \cdot \tilde{p}_{t,\gamma}) = \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \cdot \exp(-\frac{1}{\gamma} \cdot \tilde{p}_{t,\gamma}) \\
&\geq \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \cdot (1 - \frac{1}{\gamma} \cdot \tilde{p}_{t,\gamma}) \\
&= \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) - \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \cdot \frac{1}{\gamma} \cdot \tilde{p}_{t,\gamma} \\
&\geq \frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(0, b] - \frac{1}{\gamma} \cdot \tilde{p}_{t,\gamma} = \frac{1}{\gamma} \cdot \mu_{t,\gamma}(0, b]
\end{aligned}$$

where the second inequality holds by noting that $\exp(-x) \geq 1 - x$ for any $x \geq 0$, and the last inequality holds from the induction condition on $t-1$ and $\exp(-x) \leq 1$ when $x \geq 0$. Thus, our proof for the first case is completed.

Case 2: When $\eta_{t,\gamma} = 0$ and $d_t \leq b$, then $(\gamma - \mu_{t-1,\gamma}(0, 1-d_t]) \cdot p_t$ measure of probability mass of $\tilde{X}_{t-1,\gamma}$ is moved from 0 into the interval $(0, b]$ in the update (3.13). However, $\eta_{t,\gamma} = 0$ implies that

$$\mu_{t-1,\gamma}(0, 1-d_t] = P(0 < \tilde{X}_{t-1,\gamma} \leq 1 - d_t) \leq \gamma$$

Since $1 - d_t \geq 1 - b$, we must have $\mu_{t-1,\gamma}(0, 1-b] \leq \mu_{t-1,\gamma}(0, 1-d_t]$. Thus, it holds that

$$\mu_{t,\gamma}(0, 1-b] \leq \mu_{t-1,\gamma}(0, 1-b] + (\gamma - \mu_{t-1,\gamma}(0, 1-d_t]) \cdot p_t \leq \mu_{t-1,\gamma}(0, 1-d_t] \cdot (1 - p_t) + \gamma \cdot p_t \leq \gamma$$

Thus, from the inequality $\exp(-x) \geq 1 - x$ for any $x \geq 0$, it holds that

$$\exp(-\frac{1}{\gamma} \cdot \mu_{t,\gamma}(b, 1-b]) \geq 1 - \frac{1}{\gamma} \cdot \mu_{t,\gamma}(b, 1-b] \geq \frac{1}{\gamma} \cdot (\mu_{t,\gamma}(0, 1-b] - \mu_{t,\gamma}(b, 1-b]) = \frac{1}{\gamma} \cdot \mu_{t,\gamma}(0, b]$$

which completes our proof for the second case.

Case 3: When $\eta_{t,\gamma} = 0$ and $b < d_t \leq 1 - b$, then $(\gamma - \mu_{t-1,\gamma}(0, 1-d_t]) \cdot p_t$ measure of probability mass of $\tilde{X}_{t-1,\gamma}$ is moved from 0 into the interval $(b, 1-b]$. However, it is guaranteed that

$$\mu_{t,\gamma}(b, 1-b] \leq \mu_{t-1,\gamma}(b, 1-b] + \gamma \cdot p_t$$

Moreover, since $1 - d_t \geq b$, then p_t fraction of the probability mass of $\tilde{X}_{t-1,\gamma}$ over the interval $(0, b]$ is moved out of this interval in the update (3.13). Thus, we have

$$\mu_{t,\gamma}(0, b] = \mu_{t-1,\gamma}(0, b] \cdot (1 - p_t)$$

It holds that

$$\begin{aligned}
\exp(-\frac{1}{\gamma} \cdot \mu_{t,\gamma}(b, 1-b]) &\geq \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b] - p_t) = \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \cdot \exp(-p_t) \\
&\geq \exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \cdot (1 - p_t) \\
&\geq \frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(0, b] \cdot (1 - p_t) = \frac{1}{\gamma} \cdot \mu_{t\gamma}(0, b]
\end{aligned}$$

where the second inequality holds by noting $\exp(-x) \geq 1 - x$ and the last inequality holds from the induction condition on $t-1$. Thus, our proof for the third case is completed.

Case 4: When $\eta_{t,\gamma} = 0$ and $d_t > 1 - b$, then the probability mass of $\tilde{X}_{t-1,\gamma}$ over the interval $(b, 1-b]$ remains unchanged and the probability mass of $\tilde{X}_{t-1,\gamma}$ over the interval $(0, b]$ can only become smaller in the update (3.13). Then, we have that

$$\mu_{t,\gamma}(0, b] \leq \mu_{t-1,\gamma}(0, b] \quad \text{and} \quad \mu_{t,\gamma}(b, 1-b] = \mu_{t-1,\gamma}(b, 1-b]$$

Thus, we complete our proof for this last case directly from the induction hypothesis that $\exp(-\frac{1}{\gamma} \cdot \mu_{t-1,\gamma}(b, 1-b]) \geq \frac{1}{\gamma} \cdot \mu_{t-1\gamma}(0, b]$. ∎

For a fixed t, assume that the random variable $\tilde{X}_{t,\gamma}$ is well-defined. Then, given the invariant (3.15) established in Lemma 3.4, we can lower bound the measure of zero-utilization sample paths, i.e., $P(\tilde{X}_{t,\gamma} = 0)$, from the constraint $\sum_{t' \leq t} p_{t'} d_{t'} \leq 1$. Note that invariant (3.15) enables us a way to upper bound the measure of "bad" sample paths with utilization within $(0, b]$ by that with utilization within $(b, 1-b]$ for some $0 < b < \frac{1}{2}$. Therefore on the remaining sample paths the utilization must be either 0 or greater than $1 - b$, which are good cases for us. From this, we show that a γ as large as $\frac{1}{3+e^{-2}} \approx 0.319$ allows for a γ-measure of sample paths to have zero utilization at time t, which implies that the random variable $\tilde{X}_{t+1,\gamma}$ is well-defined. We iteratively apply the above arguments for each $t = 1$ up to $t = T$, hence we prove the feasibility of our Best-fit Magician policy. The above arguments are formalized in the following theorem.

THEOREM 3.2. *When $\gamma = \frac{1}{3+e^{-2}}$, the threshold policy π_γ is feasible and has a competitive ratio at least $\frac{1}{3+e^{-2}}$.*

Proof. It is enough to prove that the threshold policy π_γ is feasible when $\gamma = \frac{1}{3+e^{-2}}$. In the remaining proof, we set $\gamma = \frac{1}{3+e^{-2}}$. For a fixed t, and any a and b, denote $\mu_{t,\gamma}(a, b] = P(a < \tilde{X}_{t,\gamma} \leq b)$ assuming $\tilde{X}_{t,\gamma}$ is well-defined, it is enough to prove that $\mu_{t,\gamma}(0, 1] \leq 1 - \gamma$ thus the random variable $\tilde{X}_{t+1,\gamma}$ is well-defined.

We define $U_t(s) = \mu_{t,\gamma}(0, s]$ for any $s \in (0, 1]$. Note that by definition, we have $\mathbb{E}[\tilde{X}_{t,\gamma}] = \gamma \cdot \sum_{\tau=1}^{t} p_\tau \cdot d_\tau \leq \gamma$. From integration by parts, we have that

$$\gamma \geq \mathbb{E}[\tilde{X}_{t,\gamma}] = \int_{s=0}^{1} s dU_t(s) = U_t(1) - \int_{s=0}^{1} U_t(s) ds \tag{3.16}$$

We then bound the term $\int_{s=0}^{1} U_t(s) ds$. Now suppose $U_t(1) \geq \gamma$, otherwise $U_t(1) < \gamma$ immediately implies that $U_t(1) \leq 1 - \gamma$, which proves our result. Then there must exists a constant $u^* \in (0, 1)$ such that $\gamma \cdot u^* - \gamma \cdot \ln(u^*) = U_t(1)$. We further define

$$s^* = \begin{cases} \min\{s \in (0, 1/2] : U_t(s) \geq \gamma \cdot u^*\}, & \text{if } U_t(\frac{1}{2}) \geq \gamma \cdot u^* \\ \frac{1}{2}, & \text{if } U_t(\frac{1}{2}) < \gamma \cdot u^* \end{cases}$$

Denote $U_t(s^*-) = \lim_{s \to s^*} U_t(s)$, it holds that

$$\begin{aligned} \int_{s=0}^{1} U_t(s) ds &= \int_{s=0}^{s^*-} U_t(s) ds + \int_{s=s^*}^{1/2} U_t(s) ds + \int_{s=1/2}^{1-s^*} U_t(s) ds + \int_{s=1-s^*}^{1} U_t(s) ds \\ &\leq s^* \cdot (U_t(s^*-) + U_t(1)) + \int_{s=s^*}^{1/2} U_t(s) ds + \int_{s=1/2}^{1-s^*} U_t(s) ds \\ &\leq s^* \cdot (2\gamma u^* - \gamma \cdot \ln(u^*)) + \int_{s=s^*}^{1/2} (2U_t(s) - \gamma \cdot \ln(\frac{U_t(s)}{\gamma})) ds \end{aligned}$$

where the last inequality holds by noting that $U_t(s^*-) \leq \gamma u^*$ and for any $s \in [s^*, 1/2]$, from Lemma 3.4, we have $\frac{U_t(s)}{\gamma} \leq \exp(-\frac{U_t(1-s) - U_t(s)}{\gamma})$, which implies that $\frac{U_t(1-s)}{\gamma} \leq \frac{U_t(s)}{\gamma} - \ln(\frac{U_t(s)}{\gamma})$. Note that for any $s \in [s^*, 1/2]$, we have that $\gamma \cdot u^* \leq U_t(s^*) \leq U_t(s) \leq U_t(1/2) \leq \gamma$, where $U_t(1/2) \leq \gamma$ holds directly from Lemma 3.4. Further note that the function $2x - \gamma \cdot \ln(x/\gamma)$ is a convex function, thus is quasi convex. Then, for any $s \in [s^*, 1/2]$, it holds that

$$2U_t(s) - \gamma \cdot \ln(\frac{U_t(s)}{\gamma}) \leq \max\{2\gamma u^* - \gamma \cdot \ln(u^*), 2\gamma\}$$

Thus, we have that

$$\int_{s=0}^{1} U_t(s)ds \leq s^* \cdot (2\gamma u^* - \gamma \cdot \ln(u^*)) + (1/2 - s^*) \cdot \max\{2\gamma u^* - \gamma \cdot \ln(u^*), 2\gamma\}$$

If $2\gamma u^* - \gamma \cdot \ln(u^*) \leq 2\gamma$, we have $\int_{s=0}^{1} U_t(s)ds \leq 2s^*\gamma + \gamma - 2s^*\gamma = \gamma$. From (3.16), we have that $U_t(1) \leq 2\gamma < 1-\gamma$. If $2\gamma u^* - \gamma \cdot \ln(u^*) > 2\gamma$, we have $\int_{s=0}^{1} U_t(s)ds \leq \gamma u^* - \frac{\gamma}{2} \cdot \ln(u^*)$. From (3.16) and the definition of u^*, we have that

$$U_t(1) = \gamma u^* - \gamma \cdot \ln(u^*) \leq \gamma + \gamma u^* - \frac{\gamma}{2} \cdot \ln(u^*)$$

which implies that $u^* \geq \exp(-2)$. Note that the function $x - \ln(x)$ is non-increasing on $(0, 1)$, we have $U_t(1) \leq \gamma \cdot \exp(-2) + 2\gamma = 1 - \gamma$, which completes our proof. □

It is interesting to see the contrast between our Best-fit Magician policy and the sand process defined in [4] used to prove a $1 - 1/\sqrt{k}$-competitive ratio for online generalized assignment problem, where the maximum size of each query is at most $1/k$. As illustrated in Figure 2, our policy tries to pack *as tightly as possible* by inserting the new query to the right of the existing queries, while the sand process in [4] first fits into empty rows to keep the occupied length of each row as *small* as possible. Specifically, [4] defines a threshold for each t and accepts the query (possibly randomly) as long as the occupied capacity is *smaller* than this threshold, while our policy defines a different threshold and accepts the query as long as the occupied capacity is *larger* than our threshold. Although our results can be generalized to random sizes, we need to assume that size is observed *before* the algorithm makes a decision, whereas in [4] size is randomly realized *after* the algorithm decides to serve a query. This distinction allows our problem to have a constant-factor guarantee that holds even when queries can have size 1.

4 Multi-unit Prophet Inequalities

We now consider a special case of our problem where $d_t = \frac{1}{k}$ for an integer k, for each t, i.e., the online decision maker can serve at most k queries to collect the corresponding rewards. Note that when $k = 1$, our problem reduces to the well-known prophet inequality [21], and for general k, we get the so-called multi-unit prophet inequalities [2]. The main result of this section is that for each k, we derive the tight competitive ratio for the k-unit prophet inequality problem with respect to the LP upper bound, or equivalently the optimal solution γ_k^* to the k-unit OCRS problem. Note that our values γ_k^* strictly exceed $1 - \frac{1}{\sqrt{k+3}}$ for all $k > 1$, and hence we also improve the best-known prophet inequalities for all $k > 1$.

The structure of our proof follows the three steps outlined in Subsection 1.1. To help the reader understand our main techniques, we will keep a running example from the $k = 2$ case. Here, we include the complete proof for $k = 2$ and the statements for general k. We defer the formal proof for general k to the full version [19].

4.1 LP formulation of k-unit OCRS problem. We now present our LP formulation of the k-unit OCRS problem, assuming that the vector $\boldsymbol{p}$ is given in advance.

$$\text{(4.17)} \qquad \text{Dual}(\boldsymbol{p}, k) = \max \quad \theta$$

$$\text{(4.17a)} \qquad \text{s.t.} \quad \theta \leq \frac{\sum_{l=1}^{k} x_{l,t}}{p_t} \quad \forall t$$

$$\text{(4.17b)} \qquad x_{1,t} \leq p_t \cdot (1 - \sum_{\tau<t} x_{1,\tau}) \quad \forall t$$

$$\text{(4.17c)} \qquad x_{l,t} \leq p_t \cdot \sum_{\tau<t} (x_{l-1,\tau} - x_{l,\tau}) \quad \forall t, \forall l = 2, \ldots, k$$

$$x_{1,t} \geq 0, x_{2,t} \geq 0, \ldots, x_{k,t} \geq 0$$

Here, the variable θ can be interpreted as guarantee γ in the k-unit OCRS problem and $x_{l,t}$ can be interpreted as the ex-ante probability of serving query t as the l-th one. Then, constraint (4.17a) guarantees that each query t is served with an ex-ante probability $\theta \cdot p_t$. Moreover, it is easy to see that the term $\sum_{\tau<t} x_{l-1,\tau}$ can be interpreted as the probability that the number of served queries has "reached" $l-1$ during the first $t-1$ periods, while the term $\sum_{\tau<t} x_{l,\tau}$ can be interpreted as the probability that the number of served queries is larger than $l-1$. Then, the term $\sum_{\tau<t}(x_{l-1,\tau} - x_{l,\tau})$ denotes the probability that the number of served queries is $l-1$ at the beginning of period t. Similarly, the term $1 - \sum_{\tau<t} x_{1,\tau}$ denotes the probability that no query is served at the beginning of period t. Further note that each query t can be served only after it becomes active, which happens independently with probability p_t, we get constraint (4.17b) and (4.17c). It is also direct to see that $\mathrm{Dual}(\boldsymbol{p}, k)$ is simply a special case of $\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$ in (3.9). Then, Lemma 3.3 implies that it is enough to consider $\mathrm{Dual}(\boldsymbol{p}, k)$ to obtain the tight guarantee for the k-unit prophet inequality problem.

Although presented in a different context, the "γ-Conservative Magician" procedure of [2] implies a feasible solution to $\mathrm{Dual}(\boldsymbol{p}, k)$, for any $\boldsymbol{p}$ and any k satisfying $\sum_{t=1}^{T} p_t \le k$. We now describe this implied solution in Definition 4.1, which is based on a pre-determined θ.

Definition 4.1. *A candidate solution to Dual($\boldsymbol{p}, k$) in (4.17) given θ (potentially infeasible)*

1. *For a fixed $\theta \in [0,1]$, we define $x_{1,t}(\theta) = \theta \cdot p_t$ from $t=1$ up to $t=t_2$, where t_2 is defined as the first time among $\{1,\dots,T\}$ such that $\theta > 1 - \sum_{t=1}^{t_2} \theta \cdot p_t$ and if such t_2 does not exists, we denote $t_2 = T$. Then we define that $x_{1,t}(\theta) = p_t \cdot (1 - \sum_{\tau=1}^{t-1} x_{1,\tau}(\theta))$ from $t = t_2+1$ up to $t=T$.*

2. *For $l = 2, 3, \dots, k-1$, do the following:*

   (a) *Define $x_{l,t}(\theta) = 0$ from $t=1$ up to $t=t_l$.*

   (b) *Define $x_{l,t}(\theta) = \theta \cdot p_t - \sum_{v=1}^{l-1} x_{v,t}(\theta)$ from $t = t_l+1$ up to $t = t_{l+1}$, where t_{l+1} is defined as the first time among $\{1,\dots,T\}$ such that*

   $$\theta \cdot p_{t_{l+1}+1} - \sum_{v=1}^{l-1} x_{v,t_{l+1}+1}(\theta) > p_{t_{l+1}+1} \cdot \sum_{t=1}^{t_{l+1}} (x_{l-1,t}(\theta) - x_{l,t}(\theta))$$

   and if such t_{l+1} does not exists, we denote $t_{l+1} = T$.

   (c) *Define $x_{l,t}(\theta) = p_t \cdot \sum_{\tau=1}^{t-1}(x_{l-1,\tau}(\theta) - x_{l,\tau}(\theta))$ from $t = t_{l+1}+1$ up to $t=T$.*

3. *Define $x_{k,t}(\theta) = 0$ from $t=1$ up to $t=t_k$ and define $x_{k,t}(\theta) = \theta \cdot p_t - \sum_{v=1}^{k-1} x_{v,t}(\theta)$ from $t = t_k+1$ up to $t=T$.*

Note that in the above construction, the values of $\{t_l\}$ and $\{x_{l,t}(\theta)\}$ are uniquely determined by θ. Obviously, for an arbitrary θ, the solution $\{\theta, x_{l,t}(\theta)\}$ is not necessarily a feasible solution to $\mathrm{Dual}(\boldsymbol{p}, k)$, not to mention optimality. [3] shows that if we set $\theta = 1 - \frac{1}{\sqrt{k+3}}$, then $\{\theta, x_{l,t}(\theta)\}$ is a feasible solution to $\mathrm{Dual}(\boldsymbol{p}, k)$. Thus, they obtain a $1 - \frac{1}{\sqrt{k+3}}$-lower bound of the optimal competitive ratio. However, we now try to identify a θ^*, which is *dependent on* $\boldsymbol{p}$, such that $\{\theta^*, x_{l,t}(\theta^*)\}$ is an *optimal* solution to $\mathrm{Dual}(\boldsymbol{p}, k)$.

Remark. In Definition 4.1 we describe the LP solution instead of the corresponding policy, because our proof of optimality is *implicit*, relying on duality and complementary slackness to eventually show that our derived bound of γ_k^* (which does not have a closed form) is tight. By contrast, earlier for knapsack (see Algorithm 1) we described the corresponding policy, since we were able to *explicitly* establish the bound $\frac{1}{3+e^{-2}}$ and show that it is tight through a counterexample. Although here we are able to construct counterexamples as well, this will be done implicitly through LP duality.

To aid the reader, we provide an example of the construction from Definition 4.1 with $k=2$.

Example from $k = 2$ case. We first write the simpler form of Dual($\boldsymbol{p}, k$) when $k = 2$:

$$\text{Dual}(\boldsymbol{p}, 2) = \max \quad \theta \tag{4.18}$$

$$\text{s.t.} \quad \theta \le \frac{x_{1,t} + x_{2,t}}{p_t} \quad \forall t \tag{4.18a}$$

$$x_{1,t} \le p_t \cdot (1 - \sum_{\tau<t} x_{1,\tau}) \quad \forall t \tag{4.18b}$$

$$x_{2,t} \le p_t \cdot \sum_{\tau<t} (x_{1,\tau} - x_{2,\tau}) \quad \forall t, \tag{4.18c}$$

$$x_{1,t} \ge 0, x_{2,t} \ge 0$$

Take the example where $T = 3$ and $p_1 = p_2 = p_3 = \frac{2}{3}$. For each fixed θ, we constructed $\{x_{l,t}\}$ such that $\theta \cdot p_t = \sum_{l=1}^{2} x_{l,t}$ for each $t = 1, 2, 3$. We set $x_{l,t} = 0$. Then, for each t, the value of $x_{1,t}$ will first be increased until constraint (4.18b) is binding and then the value of $x_{2,t}$ is increased. Specifically, suppose that the value of θ is fixed to be $\frac{15}{19}$, when $t = 1$, the value of $x_{1,1}$ is first increased to $\theta \cdot p_1 = \frac{10}{19}$ such that constraint (4.18a) is binding and constraint (4.18b), (4.18c) are feasible. When $t = 2$, the value of $x_{1,2}$ is first increased, however, it can at most be increased to $\frac{6}{19}$ then constraint (4.18b) will be binding. Then the value of $x_{2,2}$ is increased to $\frac{4}{19}$ such that (4.18a) is binding while (4.18c) is feasible. Finally, when $t = 3$, the value of $x_{1,3}$ is first increased to $\frac{2}{19}$ such that (4.18b) is binding. Then $x_{2,3}$ is set to be $\frac{8}{19}$ to make (4.18a) binding. Note that in this case, constraint (4.18c) for $t = 3$ also becomes binding when $\theta = \frac{15}{19}$. □

4.2 Characterizing the optimal LP solution for a given p. We begin with the following condition on θ for $\{\theta, x_{l,t}(\theta)\}$ to be feasible to Dual($\boldsymbol{p}, k$).

LEMMA 4.1. *For any vector $\boldsymbol{p}$, there exists a unique $\theta^* \in [0, 1]$ such that $\sum_{\tau=1}^{T-1} x_{k,\tau}(\theta^*) = 1 - \theta^*$. Moreover, for any $\theta \in [0, \theta^*]$, $\{\theta, x_{l,t}(\theta)\}$ is a feasible solution to Dual($\boldsymbol{p}, k$).*

We now prove that $\{\theta^*, x_{l,t}(\theta^*)\}$ is an optimal solution to Dual($\boldsymbol{p}, k$). The dual of Dual($\boldsymbol{p}, k$) can be formulated as follows:

$$\begin{aligned}
\text{Primal}(\boldsymbol{p}, k) \;=\; \min \quad & \sum_{t=1}^{T} p_t \cdot \beta_{1,t} \\
\text{s.t.} \quad & \beta_{l,t} + \sum_{\tau>t} p_\tau \cdot (\beta_{l,\tau} - \beta_{l+1,\tau}) - \xi_t \ge 0, \quad \forall t = 1, \ldots, T, \ \forall l = 1, 2, \ldots, k-1 \\
& \beta_{k,t} + \sum_{\tau>t} p_\tau \cdot \beta_{k,\tau} - \xi_t \ge 0, \quad \forall t = 1, \ldots, T \\
& \sum_{t=1}^{T} p_t \cdot \xi_t = 1 \\
& \beta_{l,t} \ge 0, \xi_t \ge 0, \quad \forall t = 1, \ldots, T, \forall l = 1, \ldots, k
\end{aligned} \tag{4.19}$$

The above LP (4.19) is in fact a re-formulation of Primal($\boldsymbol{p}, \boldsymbol{D}$) in (3.8), which is derived from DP formulation, by the following variables substitution: $\beta_{l,t} = V_t(1 - \frac{l-1}{k}) - V_{t+1}(1 - \frac{l-1}{k})$ for each $l = 1, \ldots, k$ and $\xi_t = \hat{r}_t$ for each t. Then the variable $W_t(c_t)$ can be canceled by summing up constraint (3.8a) and constraint (3.8b), while the constraint $W_t(c_t) \ge 0$ for all c_t is equivalent to $\beta_{1,t}$ and $\beta_{2,t}$ are nonnegative. Thus, for each l and t, the variable $\beta_{l,t}$ can be interpreted as the marginal increase of the total collected reward over time, when there are $l - 1$ queries being served prior to time t, and variable ξ_t can be interpreted as the reward of query t in the worst case.

In order to prove the optimality of $\{\theta^*, x_{l,t}(\theta^*)\}$, we will construct a feasible dual solution $\{\beta^*_{l,t}, \xi^*_t\}$ to Primal($\boldsymbol{p}, k$) such that *complementary slackness conditions* holds for the primal-dual pair $\{\theta^*, x_{l,t}(\theta^*)\}$ and $\{\beta^*_{l,t}, \xi^*_t\}$, then, the well-known primal/dual optimality criteria [12] establishes that $\{\theta^*, x_{l,t}(\theta^*)\}$ and $\{\beta^*_{l,t}, \xi^*_t\}$ are the optimal primal-dual pair to Dual($\boldsymbol{p}, k$) and Primal($\boldsymbol{p}, k$), which completes our proof. The above arguments are formalized in the following theorem.

THEOREM 4.1. *The solution* $\{\theta^*, x_{l,t}(\theta^*)\}$ *is optimal to Dual*$(\boldsymbol{p}, k)$*, where* θ^* *is the unique solution to* $\sum_{\tau=1}^{T-1} x_{k,\tau}(\theta^*) = 1 - \theta^*$.

Theorem 4.1 shows that Definition 4.1 constructs an optimal solution to Dual$(\boldsymbol{p}, k)$, *as long as it is given the optimal* θ^*, as defined in Lemma 4.1. This optimal θ^* is uniquely defined based on $\boldsymbol{p}$. Lemma 4.1 further shows that any $\theta \le \theta^*$ is feasible, and hence if we can find a θ which is no greater than the θ^* arising from any $\boldsymbol{p}$, then Definition 4.1 will correspond to a $\boldsymbol{p}$-agnostic procedure for the k-unit prophet inequality or OCRS problem with a guarantee of θ.

To aid the reader, we provide a complete construction for Theorem 4.1 in the $k = 2$ case below.

$k = 2$ **case.** We take the dual of the Dual$(\boldsymbol{p}, 2)$ in (4.18) and obtain the following LP:

$$
\begin{aligned}
\text{Primal}(\boldsymbol{p}, 2) \;=\; \min \;& \sum_{t=1}^{T} p_t \cdot \beta_{1,t} \\
\text{s.t. } & \beta_{1,t} + \sum_{\tau>t}^{T} p_\tau \cdot (\beta_{1,\tau} - \beta_{2,\tau}) - \xi_t \ge 0, \quad \forall t = 1, \dots, T, \\
& \beta_{2,t} + \sum_{\tau>t}^{T} p_\tau \cdot \beta_{2,\tau} - \xi_t \ge 0, \quad \forall t = 1, \dots, T \\
& \sum_{t=1}^{T} p_t \cdot \xi_t = 1 \\
& \beta_{1,t} \ge 0, \beta_{2,t} \ge 0, \xi_t \ge 0, \quad \forall t = 1, \dots, T,
\end{aligned} \tag{4.20}
$$

As illustrated above, the feasible solution from Definition 4.1 is based on a fixed θ, and defines $x_{1,t}(\theta) = \theta \cdot p_t$ for $t = 1$ up to $t = t_2$, where t_2 is the first time among $\{1, \dots, T\}$ such that $\theta > 1 - \sum_{t=1}^{t_2} \theta \cdot p_t$; if such a t_2 doesn't exist, we simply denote $t_2 = T$. Then we define $x_{1,t}(\theta) = p_t \cdot (1 - \sum_{\tau=1}^{t-1} x_{1,\tau}(\theta))$ for $t = t_2 + 1$ up to $t = T$ and define $x_{2,t}(\theta) = 0$ for $t = 1$ up to $t = t_2$, define $x_{2,t}(\theta) = \theta \cdot p_t - x_{1,t}(\theta)$ for $t = t_2 + 1$ up to $t = T$. The result from [3] implies that if we take $\theta = 1 - \frac{1}{\sqrt{k+3}} = 1 - \frac{1}{\sqrt{5}}$, then $\{\theta, x_{l,t}(\theta)\}$ forms a feasible solution to Dual$(\boldsymbol{p}, 2)$, and thus $1 - \frac{1}{\sqrt{5}}$ is a lower bound on the optimal ratio. However, we shall identify the optimal θ^*, which can depend on $\boldsymbol{p}$, such that $\{\theta^*, x_{l,t}(\theta^*)\}$ is an optimal solution to Dual$(\boldsymbol{p}, 2)$, which enables us to find the optimal ratio.

Now for a fixed $\boldsymbol{p}$ satisfying $\sum_{t=1}^{T} p_t = 2$, we can prove that there exists a unique solution to the equation $\sum_{\tau=1}^{T-1} x_{2,\tau}(\theta) = 1 - \theta$, which is equivalent to $x_{2,T}(\theta) = p_t \cdot \sum_{t=1}^{T-1}(x_{1,t}(\theta) - x_{2,t}(\theta))$, and we denote θ^* as this solution. One can verify from definitions that $\{\theta^*, x_{l,t}(\theta^*)\}$ is feasible to Dual$(\boldsymbol{p}, 2)$. We then prove its optimality by constructing a feasible solution to Primal$(\boldsymbol{p}, 2)$, denoted as $\{\beta^*_{l,t}, \xi^*_t\}$, while satisfying the *complementary slackness conditions*. Specifically, in addition to feasibility, $\{\beta^*_{l,t}, \xi^*_t\}$ needs to satisfy the following equations for each $t = 1, \dots, T$:

$$
\begin{aligned}
& \beta^*_{1,t} \cdot \left(x_{1,t}(\theta^*) - p_t \cdot (1 - \sum_{\tau<t} x_{1,\tau}(\theta^*)) \right) = \beta^*_{2,t} \cdot \left(x_{2,t}(\theta^*) - p_t \cdot \sum_{\tau<t} (x_{1,\tau}(\theta^*) - x_{2,\tau}(\theta^*)) \right) = 0, \\
& x_{1,t}(\theta^*) \cdot \left(\beta^*_{1,t} + \sum_{\tau>t}^{T} p_\tau \cdot (\beta^*_{1,\tau} - \beta^*_{2,\tau}) - \xi^*_t \right) = x_{2,t}(\theta^*) \cdot \left(\beta^*_{2,t} + \sum_{\tau>t}^{T} p_\tau \cdot \beta^*_{2,\tau} - \xi^*_t \right) = 0,
\end{aligned} \tag{4.21}
$$

Then, from the well-known primal/dual optimality criteria [12], the primal-dual pair $\{\theta^*, x_{l,t}(\theta^*)\}$ and $\{\beta^*_{l,t}, \xi^*_t\}$ should be optimal. Note that from definitions, $\{\theta^*, x_{l,t}(\theta^*)\}$ satisfies the following conditions:

$$
\begin{aligned}
& x_{1,t}(\theta^*) \le p_t \cdot (1 - \sum_{\tau<t} x_{1,\tau}(\theta^*)), \ \forall t \le t_2 && x_{1,t}(\theta^*) > 0, \ \forall t = 1, \dots, T \\
& x_{2,t}(\theta^*) \le p_t \cdot \sum_{\tau<t} (x_{1,\tau}(\theta^*) - x_{2,\tau}(\theta^*)), \ \forall t \le T - 1 && x_{2,t}(\theta^*) > 0, \ \forall t \ge t_2 + 1
\end{aligned}
$$

where t_2 is the time index associated with the definition of $\{x_{l,t}(\theta^*)\}$. Thus, in order for $\{\beta^*_{l,t}, \xi^*_t\}$ to satisfy the conditions (4.21), it is enough for $\{\beta^*_{l,t}, \xi^*_t\}$ to satisfy the following equations for each $t = 1, \ldots, T$:

$$\beta^*_{1,t} = 0,\ \forall t \le t_2, \qquad \beta^*_{1,t} + \sum_{\tau > t}^{T} p_\tau \cdot (\beta^*_{1,\tau} - \beta^*_{2,\tau}) - \xi^*_t = 0,\ \forall t = 1, \ldots, T \tag{4.22}$$

$$\beta^*_{2,t} = 0,\ \forall t \le T-1, \qquad \beta^*_{2,t} + \sum_{\tau > t}^{T} p_\tau \cdot \beta^*_{2,\tau} - \xi^*_t = 0,\ \forall t \ge t_2 + 1 \tag{4.23}$$

A key observation is that given the fixed time index t_2, the values of $\{\beta^*_{l,t}, \xi^*_t\}$ can be automatically determined from the above conditions. Specifically, we must have $\beta^*_{1,T} = \beta^*_{2,T} = \xi^*_T = R$, where R is a constant to be specified later. Then, from condition (4.23) holds for $t \ge t_2 + 1$, together with $\beta^*_{2,t} = 0$ for $t \le T-1$, we must have $\xi^*_t = p_T R$ for $t_2 + 1 \le t \le T-1$. Also, we know from condition (4.22) and $\beta^*_{2,t} = 0$ for $t \le T-1$ that

$$\beta^*_{1,t} + \sum_{\tau > t}^{T-1} p_\tau \cdot \beta^*_{1,\tau} = \xi^*_t = p_T R \ \text{ holds for all } t_2 + 1 \le t \le T-1$$

which enables us to compute iteratively that $\beta^*_{1,t} = \prod_{\tau=t+1}^{T-1}(1 - p_\tau) \cdot p_T R$ for $t_2 + 1 \le t \le T-1$. Finally, for $t \le t_2$, we have

$$\xi^*_t = \beta^*_{1,t} + \sum_{\tau=t+1}^{T-1} p_\tau \cdot \beta^*_{1,\tau} = \sum_{\tau=t_2+1}^{T-1} p_\tau \cdot \beta^*_{1,\tau} = \left(1 - \prod_{\tau=t_2+1}^{T-1}(1 - p_\tau)\right) \cdot p_T R$$

The constant R is determined such that $\sum_{t=1}^{T} p_t \cdot \xi^*_t = 1$, which must be positive. In this way, we derive the closed-form expression of $\{\beta^*_{l,t}, \xi^*_t\}$, which satisfies conditions (4.22) and (4.23). It is also direct to check that the constructed $\{\beta^*_{l,t}, \xi^*_t\}$ is feasible by noting that $\beta^*_{2,t} + \sum_{\tau > t}^{T} p_\tau \cdot \beta^*_{2,\tau} = p_T R \ge \xi^*_t$ for all $t \le t_2$. Thus, we obtain the optimal primal-dual pair $\{\theta^*, x_{l,t}(\theta^*)\}$ and $\{\beta^*_{l,t}, \xi^*_t\}$. $\square$

4.3 Characterizing the worst-case distribution. Our goal is now to find the $\boldsymbol{p}$ such that the optimal objective value θ^* of Dual$(\boldsymbol{p}, k)$ in (4.17) reaches its minimum. We would like to characterize the worst-case distribution and then compute the competitive ratio.

We first characterize the worst case distribution on which the optimal objective value of Dual$(\boldsymbol{p}, k)$ reaches its minimum. Obviously, it is enough for us to consider only the $\boldsymbol{p}$ satisfying $\sum_{t=1}^{T} p_t = k$. We show in the following lemma that splitting one query into two queries can only make the optimal objective value of Dual$(\boldsymbol{p}, k)$ become smaller, and thus, in the worst case distribution, each p_t should be infinitesimal small.

LEMMA 4.2. *For any $\boldsymbol{p} = (p_1, \ldots, p_T)$ satisfying $\sum_{t=1}^{T} p_t = k$, and any $\sigma \in [0, 1]$, $1 \le q \le T$, if we define a new sequence of arrival probabilities $\tilde{\boldsymbol{p}} = (\tilde{p}_1, \ldots, \tilde{p}_{T+1})$ such that*

$$\tilde{p}_t = p_t \quad \forall t < q, \qquad \tilde{p}_q = p_q \cdot \sigma, \qquad \tilde{p}_{q+1} = p_q \cdot (1 - \sigma) \ \ \text{and} \ \ \tilde{p}_{t+1} = p_t \quad \forall q + 1 \le t \le T$$

then it holds that Dual$(\boldsymbol{p}, k) \ge$ Dual$(\tilde{\boldsymbol{p}}, k)$.

Now, for each $\boldsymbol{p} = (p_1, \ldots, p_T)$ satisfying $\sum_{t=1}^{T} p_t = k$, we assume without loss of generality that p_t is a rational number for each t, i.e., $p_t = \frac{n_t}{N}$ where n_t is an integer for each t and N is an integer to denote the common denominator. We first split p_1 into $\frac{1}{N}$ and $\frac{n_1 - 1}{N}$ to form a new sequence of arrival probabilities. From Lemma 4.2, we know such an operation can only decrease the optimal objective value of Dual$(\boldsymbol{p}, k)$. We then split $\frac{n_1-1}{N}$ into $\frac{1}{N}$ and $\frac{n_1-2}{N}$ and so on. In this way, we split p_1 into n_1 copies of $\frac{1}{N}$ to form a new sequence of arrival probabilities and Lemma 4.2 guarantees that the optimal objective value of Dual$(\boldsymbol{p}, k)$ can only become smaller. We repeat the above operation for each t. Finally, we form a new sequence of arrival probabilities, denoted as $\boldsymbol{p}^N = (\frac{1}{N}, \ldots, \frac{1}{N}) \in \mathbb{R}^{Nk}$, and we have Dual$(\boldsymbol{p}, k) \ge$ Dual$(\boldsymbol{p}^N, k)$. Intuitively, when $N \to \infty$, the optimal objective value of Dual$(\boldsymbol{p}, k)$ will reach its minimum. Note that when $N \to \infty$, we always have $\sum_{t=1}^{Nk} p^N_t = k$, then the arrival Bernoulli process will approximate a Poisson process with rate 1 over the time interval $[0, k]$. The above arguments implies that the worst-case arrival process is a Poisson process.

Under the Poisson process, for each fixed ratio $\theta \in [0, 1]$, our solution in Definition 4.1 can be interpreted as a solution to an ordinary differential equation (ODE). We further note that for $\boldsymbol{p}^N$ and any $\theta \in [0, 1]$, our solution in Definition 4.1 can be regarded as the solution obtained from applying Euler's method to solve this ODE by uniformly discretizing the interval $[0, k]$ into Nk discrete points. Then, for any fixed ratio $\theta \in [0, 1]$, after showing the Lipshictz continuity of the function defining this ODE, we can apply the global truncation error theorem of Euler's method (Theorem 212A [8]) to establish the solution under Poisson process as the limit of the solution under $\boldsymbol{p}^N$ when $N \to \infty$. Based on this convergence, we can prove that the optimal value under the Poisson process is equivalent to $\lim_{N\to\infty} \mathrm{Dual}(\boldsymbol{p}^N, k)$, which is the optimal ratio we are looking for.

$k = 2$ **case.** Given the closed-form expressions we have obtained above, we can show that the value of $\mathrm{Dual}(\boldsymbol{p}, 2)$ reaches its infimum as $T \to \infty$ and each p_t becomes infinitesimal. Note that we always have the equation $\sum_{t=1}^{T} p_t = 2$. As $T \to \infty$ and each p_t becomes infinitesimal, the arrival Bernoulli process will finally approximate a Poisson process with rate 1 over the time interval $[0, 2]$. Thus, we characterize the Poisson process as the worst-case distribution. In this case, when $\theta \le \frac{1}{2}$, the value of $\sum_{t=1}^{T-1} x_{2,t}(\theta)$ is always 0 and when $\theta > \frac{1}{2}$, the value of $\sum_{t=1}^{T-1} x_{2,t}(\theta)$ will approximate $-1 + 2\theta + \theta \cdot \exp(\frac{1-\theta}{\theta} - 2)$. Solving the equation

$$-1 + 2\theta + \theta \cdot \exp(\frac{1-\theta}{\theta} - 2) = 1 - \theta$$

for $\theta > \frac{1}{2}$, we get $\theta \approx 0.6148$, which is the optimal competitive ratio γ_2^* for $k = 2$. $\square$

For a general k, the values of γ_k^* have previously appeared in [27] through the analysis of a "reflecting" Poisson process. However, we show that these values γ_k^* are *optimal*, deriving them instead through $\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$. Moreover, [27] assumes Poisson arrivals to begin with, while we allow for arbitrary probability vectors $\boldsymbol{p}$ and show that the limiting Poisson case is the worst case.

Specifically in the case of $k = 2$, we construct an example showing that relative to the weaker prophet benchmark $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]$, it is not possible to do much better than γ_2^*. Our construction is based on adapting the tight example relative to the stronger benchmark $\mathrm{UP}(\hat{\mathcal{H}})$, which can be found through the values of $\hat{r}_t$ in our optimal solution to the primal LP (with the values of p_t being infinitesimal). We note that this suggests there is *some separation* between optimal ex-ante vs. non-ex-ante prophet inequalities when $k > 1$, which is not the case when $k = 1$ (because they are both 1/2). The above discussion is formalized in the following proposition and the proof is relegated to Section E.

Proposition 4.1. *For the 2-unit prophet inequality problem, it holds that* $\inf_{\mathcal{H}} \frac{\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^{\pi}(\boldsymbol{I})]}{\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathit{off}}(\boldsymbol{I})]} \le 0.6269$ *for any online algorithm* π.

We now discuss how the construction in Definition 4.1 should be interpreted when the arrival process is a Poisson process. We find it is more convenient to work with the functions $\{\tilde{y}_{l,\theta}(\cdot)\}_{\forall l=1,\dots,k}$ over $[0, k]$, where $\tilde{y}_{l,\theta}(t)$ denotes the ex-ante probability that there is a query served as the l-th one during the period $[0, t]$. Note that the variable $x_{l,t}(\theta)$ denotes the ex-ante probability that there is a query accepted as the l-th query at time t, hence we have $x_{l,t}(\theta) = d\tilde{y}_{l,\theta}(t)$. We denote $\tilde{y}_{0,\theta}(t) = 1$ for each $t \in [0, k]$. Then the functions $\{\tilde{y}_{l,\theta}(\cdot)\}_{\forall l=1,\dots,k}$ corresponding to the construction in Definition 4.1 under Poisson arrivals can be interpreted as follows.

Definition 4.2. *(Ordinary Differential Equation (ODE) formula under Poisson Arrival)*

1. *For each fixed* $\theta \in [0, 1]$*, we define* $\tilde{y}_{0,\theta}(t) = 1$ *for each* $t \in [0, k]$ *and* $t_1 = 0$*.*
2. *For each* $l = 1, 2, \dots, k - 1$*, do the following:*
   (a) $\tilde{y}_{l,\theta}(t) = 0$ *when* $t \le t_l$*.*
   (b) *When* $t_l \le t \le t_{l+1}$*, it holds that*

$$\frac{d\tilde{y}_{l,\theta}(t)}{dt} = \theta - \sum_{v=1}^{l-1} \frac{d\tilde{y}_{v,\theta}(t)}{dt} = \theta - 1 + \tilde{y}_{l-1,\theta}(t), \quad \forall t_l \le t \le t_{l+1} \tag{4.24}$$

where t_{l+1} is defined as the first time that $\tilde{y}_{l,\theta}(t_{l+1}) = 1-\theta$. If such a t_{l+1} does not exists, we denote $t_{l+1} = k$.

(c) *When $t_{l+1} \le t \le k$, it holds*

$$\frac{d\tilde{y}_{l,\theta}(t)}{dt} = \tilde{y}_{l-1,\theta}(t) - \tilde{y}_{l,\theta}(t), \quad \forall t_{l+1} \le t \le k \tag{4.25}$$

3. *$\tilde{y}_{k,\theta}(t) = 0$ if $t \le t_k$ and $\frac{d\tilde{y}_{k,\theta}(t)}{dt} = \theta - 1 + \tilde{y}_{k-1,\theta}(t)$ if $t_k \le t \le k$.*

Thus, from Theorem 4.1, the solution to the equation $\tilde{y}_{k,\theta}(k) = 1-\theta$ should be the minimum of the optimal objective value of Dual$(\boldsymbol{p}, k)$ in (4.17), which is the competitive ratio γ_k^* we are seeking for. The above arguments are formalized in the following Theorem 4.2. Note that the following Theorem 4.2 is our ultimate result for the k-unit case, while Definition 4.2 characterizes the ODE formula mentioned in Subsection 1.1. In the remaining part of this section, we will describe the computational procedure for γ_k^*.

THEOREM 4.2. *For each $\theta \in [0,1]$, denote $\{\tilde{y}_{l,\theta}(\cdot)\}$ as the functions defined in Definition 4.2. Then there exists a unique $\gamma_k^* \in [0,1]$ such that $\tilde{y}_{k,\gamma_k^*}(k) = 1-\gamma_k^*$ and it holds that*

$$\gamma_k^* = \inf_{\boldsymbol{p}} \ Dual(\boldsymbol{p}, k) \quad s.t. \ \sum_{t=1}^{T} p_t = k$$

We now show that the ODE in Definition 4.2 admits an analytical solution, which enables us to compute γ_k^* for each k. For each fixed θ, when $l = 1$, it is immediate that

$$\tilde{y}_{1,\theta}(t) = \theta \cdot t, \quad \text{when } t \le t_2 = \frac{1-\theta}{\theta}, \quad \text{and} \quad \tilde{y}_{1,\theta}(t) = 1 - \theta \cdot \exp(t_2 - t), \quad \text{for } t_2 \le t \le k$$

Now suppose that there exists a fixed $2 \le l \le k$ such that for each $1 \le v \le l-1$, it holds

$$\tilde{y}_{v,\theta}(t) = \zeta_v + \theta \cdot t + \sum_{q=0}^{v-2} \zeta_{v,q} \cdot t^q \cdot \exp(-t), \qquad \text{when } t_v \le t \le t_{v+1}$$

$$\tilde{y}_{v,\theta}(t) = 1 + \sum_{q=0}^{v-1} \psi_{v,q} \cdot t^q \cdot \exp(-t), \qquad \text{when } t_{v+1} \le t \le k$$

for some parameters $\{\zeta_v, \zeta_{v,q}, \psi_{v,q}\}$, which are specified by θ. Then from ODE (4.24) and (4.25), it must hold that

$$\tilde{y}_{l,\theta}(t) = \zeta_l + \theta \cdot t + \sum_{q=0}^{l-2} \zeta_{l,q} \cdot t^q \cdot \exp(-t), \qquad \text{when } t_l \le t \le t_{l+1}$$

$$\tilde{y}_{l,\theta}(t) = 1 + \sum_{q=0}^{l-1} \psi_{l,q} \cdot t^q \cdot \exp(-t), \qquad \text{when } t_{l+1} \le t \le k$$

The parameters $\{\zeta_l, \zeta_{l,q}, \psi_{l,q}\}$ can be computed from the following steps:

1. Set $\zeta_{l,l-1} = 0$ and compute $\zeta_{l,q}$ iteratively from $q = l-2$ up to $q = 0$ by setting

$$\zeta_{l,q} = (q+1) \cdot \zeta_{l,q+1} - \psi_{l-1,q}$$

2. Set the value of ζ_l such that $\tilde{y}_{l,\theta}(t_l) = 0$. If $l = k$, we set $t_{l+1} = k$, otherwise, we set t_{l+1} to be the solution to the following equation:

$$1 - \theta = \tilde{y}_{l,\theta}(t) = \zeta_l + \theta \cdot t + \sum_{q=0}^{l-2} \zeta_{l,q} \cdot t^q \cdot \exp(-t)$$

   Note that from definition $\tilde{y}_{l,\theta}(t)$ is monotone increasing with t, thus we can do a bi-section search on the interval $[t_l, k]$ to obtain the value of t_{l+1}.

3. Set $\psi_{l,q} = \frac{\psi_{l-1,q-1}}{q}$ for each $q = 1, \ldots, l-1$. If $l < k$, the value of $\psi_{l,0}$ is determined such that

$$1 - \theta = 1 + \sum_{q=0}^{l-1} \psi_{l,q} \cdot t_{l+1}^{q} \cdot \exp(-t_{l+1})$$

Thus, for each fixed θ, we can follow the above procedure to obtain the value of $\tilde{y}_{k,\theta}(k)$. Note that the value of $\tilde{y}_{k,\theta}(k)$ is monotone increasing with θ (formally proved in Lemma 11 of full version [19]), hence we can do a bisection search on $\theta \in [0,1]$ to obtain the value of γ_k^* as the unique solution of the equation $\tilde{y}_{k,\theta}(k) = 1 - \theta$. From Theorem 4.2, γ_k^* is the optimal value for the competitive ratio.

References

[1] Adelman, D. *Dynamic bid prices in revenue management.* Operations Research. (2007)

[2] Alaei, S. *Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers.* 2011 IEEE 52nd Annual Symposium On Foundations Of Computer Science. pp. 512-521 (2011)

[3] Alaei, S., Hajiaghayi, M. & Liaghat, V. *Online prophet-inequality matching with applications to ad allocation.* Proceedings Of The 13th ACM Conference On Electronic Commerce. pp. 18-35 (2012)

[4] Alaei, S., Hajiaghayi, M. & Liaghat, V. *The online stochastic generalized assignment problem.* Approximation, Randomization, And Combinatorial Optimization. Algorithms And Techniques. pp. 11-25 (2013)

[5] Alaei, S. *Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers.* SIAM Journal On Computing. **43**, 930-972 (2014)

[6] Alaei, S., Ali M. & Azarakhsh M. *Revenue Maximization Under Unknown Private Values With Non-Obligatory Inspection.* Proceedings of the 22nd ACM Conference on Economics and Computation (2021)

[7] Beyhaghi, H., Golrezaei, N., Leme, R., Pal, M. & Sivan, B. *Improved Revenue Bounds for Posted-Price and Second-Price Mechanisms.* Operations Research. (2021)

[8] Butcher, J. & Goodwin, N. *Numerical methods for ordinary differential equations.* Wiley Online Library, 2008

[9] Chawla, S., Hartline, J., Malec, D. & Sivan, B. *Multi-parameter mechanism design and sequential posted pricing.* Proceedings Of The Forty-second ACM Symposium On Theory Of Computing. pp. 311-320 (2010)

[10] Chawla, S., Devanur, N. & Lykouris, T. *Static pricing for multi-unit prophet inequalities.* ArXiv Preprint ArXiv:2007.07990. (2020)

[11] Correa, J., Foncea, P., Hoeksma, R., Oosterwijk, T. & Vredeveld, T. *Recent developments in prophet inequalities.* ACM SIGecom Exchanges. **17**, 61-70 (2019)

[12] Dantzig, G. & Thapa, M. *Linear programming 2: theory and extensions.* Springer Science & Business Media, 2006

[13] Dutting, P., Feldman, M., Kesselheim, T. & Lucier, B. *Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs.* SIAM Journal On Computing. **49**, 540-582 (2020)

[14] Feng, Y., Niazadeh, R. & Saberi, A. *Near-Optimal Bayesian Online Assortment of Reusable Resources.* Chicago Booth Research Paper. (2020)

[15] Feldman, M., Svensson, O. & Zenklusen, R. *Online Contention Resolution Schemes with Applications to Bayesian Selection Problems.* SIAM Journal On Computing. **50**, 255-300 (2021)

[16] Garey, M., Graham, R. & Ullman, J. *Worst-case analysis of memory allocation algorithms.* Proceedings Of The Fourth Annual ACM Symposium On Theory Of Computing. pp. 143-150 (1972)

[17] Gallego, G., Li, A., Truong, V. & Wang, X. *Online resource allocation with customer choice.* ArXiv Preprint ArXiv:1511.01837. (2015)

[18] Hajiaghayi, M., Kleinberg, R. & Sandholm, T. *Automated online mechanism design and prophet inequalities.* AAAI. **7** pp. 58-65 (2007)

[19] Jiang, J., Ma, W. & Zhang, J. *Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack.* ArXiv Preprint ArXiv:2107.02058. (2021) `https://arxiv.org/abs/2107.02058`

[20] Han, X., Kawase, Y. & Makino, K. *Randomized algorithms for online knapsack problems.* Theoretical Computer Science. **562** pp. 395-405 (2015)

[21] Krengel, U. & Sucheston, L. *On semiamarts, amarts, and processes with finite value.* Probability On Banach Spaces. **4** pp. 197-266 (1978)

[22] Lee, E. & Singla, S. *Optimal online contention resolution schemes via ex-ante prophet inequalities.* ArXiv Preprint ArXiv:1806.09251. (2018)

[23] Ma, W., Simchi-Levi, D. & Zhao, J. *The Competitive Ratio of Threshold Policies for Online Unit-density Knapsack Problems.* ArXiv Preprint ArXiv:1907.08735. (2019)

[24] Ma, W., Simchi-Levi, D. & Zhao, J. *Dynamic Pricing (and Assortment) under a Static Calendar.* Management Science. **67**, 2292-2313 (2021)

[25] Papastavrou, J., Rajagopalan, S. & Kleywegt, A. *The dynamic and stochastic knapsack problem with deadlines.* Management Science. **42**, 1706-1718 (1996)

[26] Stein, C., Truong, V. & Wang, X. *Advance service reservations with heterogeneous customers.* Management Science. **66**, 2929-2950 (2020)

[27] Wang, X., Truong, V. & Bank, D. *Online advance admission scheduling for services with customer preferences.* ArXiv Preprint ArXiv:1805.10412. (2018)

[28] Yan, Q. *Mechanism design via correlation gap.* Proceedings Of The Twenty-second Annual ACM-SIAM Symposium On Discrete Algorithms. pp. 710-719 (2011)

A Proof of Lemma 3.1

Proof. Denote $\{x_t^{\text{off}}(\boldsymbol{I})\}_{t=1}^T$ as the optimal solution to $V^{\text{off}}(\boldsymbol{I})$ in (3.2) and for each t, we denote $x_t(r_t) = \mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[x_t^{\text{off}}(\boldsymbol{I})|\boldsymbol{I}_t = r_t]$, where $\boldsymbol{I}_t$ denotes the reward of query t in the sequence $\boldsymbol{I}$. Then, we must have

$$0 \le x_t(r_t) \le 1, \quad \forall t,\ \forall r_t$$

and

$$1 \ge \mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[\sum_{t=1}^T d_t \cdot x_t^{\text{off}}(\boldsymbol{I})] = \sum_{t=1}^T \mathbb{E}_{r_t\sim F_t}[\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[d_t \cdot x_t^{\text{off}}(\boldsymbol{I})|\boldsymbol{I}_t = r_t]] = \sum_{t=1}^T \mathbb{E}_{r_t\sim F_t}[d_t \cdot x_t(r_t)]$$

Thus, we conclude that $\{x_t(r_t)\}$ is a feasible solution to $\text{UP}(\mathcal{H})$ in (3.4) and it holds that

$$\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\text{off}}(\boldsymbol{I})] = \mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[\sum_{t=1}^T \boldsymbol{I}_t \cdot x_t^{\text{off}}(\boldsymbol{I})] = \sum_{t=1}^T \mathbb{E}_{r_t\sim F_t}[r_t \cdot x_t(r_t)] \le \text{UP}(\mathcal{H})$$

which completes our proof. ∎

B Proof of Lemma 3.2

Proof. For any general setup $\mathcal{H}$ and the corresponding reduced two-point distribution setup $\hat{\mathcal{H}}$, from definition, it holds that $\mathrm{UP}(\mathcal{H}) = \mathrm{UP}(\hat{\mathcal{H}})$. Thus, it is enough to construct a feasible online policy π' such that $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\pi'}(\boldsymbol{I})] = \mathbb{E}_{\hat{\boldsymbol{I}}\sim\hat{\boldsymbol{F}}}[V^{\pi}(\hat{\boldsymbol{I}})]$. Denote $\{x_t^*(r_t)\}_{t=1,\dots,T}$ as the optimal solution to $\mathrm{UP}(\mathcal{H})$, we construct the policy π' as follows:

(i). Upon the arrival of query t, where its reward is realized as r_t, we choose to present a virtual query t with reward and size $(\hat{r}_t, d_t)$ to the policy π with probability $x_t^*(r_t)$ and choose not to present the virtual query with probability $1 - x_t^*(r_t)$.

(ii). Accept item t if and only if we choose to present the virtual query t and the policy π accepts this query.

Since the policy π' accepts query t only if the virtual query t with the same size is accepted by π, we can immediately tell that the capacity constraint (3.1) is satisfied by π'. Moreover, note that from the definition of p_t in (3.5), each virtual query t is presented to the policy π with probability p_t. Thus, π is implemented on the problem setup $\hat{\mathcal{H}}$. Further note that conditioning on virtual query t is presented, the decision of policy π is independent of r_t and it holds

$$\mathbb{E}[r_t \mid \text{virtual query } t \text{ presented}] = \frac{\mathbb{E}_{r_t\sim F_t}[r_t \cdot x_t^*(r_t)]}{\mathbb{E}[1_{\{\text{virtual query } t \text{ presented}\}}]} = \hat{r}_t$$

where $1_{\{\cdot\}}$ denotes the indicator function. Thus, we have

$$\mathbb{E}[r_t \cdot x_t^{\pi'}] = \mathbb{E}\left[\mathbb{E}[r_t \mid x_t^{\pi} = 1]\right] = \mathbb{E}[\hat{r}_t \cdot x_t^{\pi}]$$

which implies that $\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\pi'}(\boldsymbol{I})] = \mathbb{E}_{\hat{\boldsymbol{I}}\sim\hat{\boldsymbol{F}}}[V^{\pi}(\hat{\boldsymbol{I}})]$. □

C Proof of Lemma 3.3

Proof. It is clear that

$$\inf_{\hat{\mathcal{H}}} \frac{V_1^*(1)}{\mathbf{UP}(\hat{\mathcal{H}})} = \inf_{(\boldsymbol{p},\boldsymbol{D}):\sum_{t=1}^T p_t\cdot d_t \le 1} \mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$$

It only remains to show that $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D}) = \mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$ for each $(\boldsymbol{p}, \boldsymbol{D})$ satisfying $\sum_{t=1}^T p_t \cdot d_t \le 1$.

Note that the size d_t is deterministic for each t, the remaining capacity c_t will only have a finite number of possible values. Thus, $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$ is essentially of finite dimension with a finite number of variables and constraints. We then introduce the dual variable $\kappa_t(c_t)$ for constraint (3.8a), the dual variable $\alpha_t(c_t)$ for constraint (3.8b) and a dual variable θ for constraint (3.8c). Then we have the following LP as the dual of $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$.

$$\begin{aligned}
\text{(C.1)} \qquad \max\ & \theta \\
\text{s.t.}\ & \theta \cdot p_t \le \sum_{c_t \ge d_t} \alpha_t(c_t), \quad \forall t \\
& \alpha_t(c_t) \le p_t \cdot \kappa_t(c_t), \quad \forall t, \forall c_t \ge d_t \\
& \kappa_t(c_t) = \kappa_{t-1}(c_t) + \alpha_{t-1}(c_t + d_{t-1}) - \alpha_{t-1}(c_t), \quad \forall t, \forall c_t \le 1 - d_t \\
& \kappa_t(c_t) = \kappa_{t-1}(c_t) - \alpha_{t-1}(c_t), \quad \forall t, \forall c_t > 1 - d_t \\
& \kappa_1(1) = 1 \\
& \theta, \alpha_t(c_t) \ge 0, \kappa_t(c_t) \ge 0, \quad \forall t, \forall c_t
\end{aligned}$$

Note that from the binding constraints, it must hold

$$\kappa_t(c_t) = \sum_{\tau \le t-1} (\alpha_{t-1}(c_t + d_{t-1}) - \alpha_{t-1}(c_t))$$

where we set $\alpha_t(c) = 0$ for all $c > 1$. Thus, we further simplify the LP (C.1) by eliminating variable $\kappa_t(c_t)$, and derive LP $\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$ in (3.9). Note that the finite dimension of $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$ implies the strong duality between $\mathrm{Primal}(\boldsymbol{p}, \boldsymbol{D})$ and $\mathrm{Dual}(\boldsymbol{p}, \boldsymbol{D})$, which completes our proof. □

D Proof of Proposition 3.1

Proof. Consider a problem setup $\hat{\mathcal{H}}$ with 4 queries and

$$(\hat{r}_1, p_1, d_1) = (r, 1, \epsilon), \quad (\hat{r}_2, p_2, d_2) = (\hat{r}_3, p_3, d_3) = (r, \frac{1-2\epsilon}{1+2\epsilon}, \frac{1}{2}+\epsilon), \quad (\hat{r}_4, p_4, d_4) = (r/\epsilon, \epsilon, 1)$$

for $r > 0$ and some $\epsilon > 0$. Obviously, if the policy π only serves queries with a size greater than 1/2, then the expected total reward is $V_L^\pi = r + O(\epsilon)$. If the policy π only serves queries with a size no greater than 1/2, then the expected total reward is $V_S^\pi = r$. Thus, the expected total reward of the policy π is

$$V^\pi = \max\{V_L^\pi, V_S^\pi\} = r + O(\epsilon)$$

Moreover, it is direct to see that $\sum_{t=1}^4 p_t \cdot d_t = 1$, then, we have $\mathrm{UP}(\hat{\mathcal{H}}) = 4r$. Thus, the guarantee of π is upper bounded by $1/4 + O(\epsilon)$, which converges to 1/4 as $\epsilon \to 0$. ∎

E Proof of Proposition 4.1

Proof. We consider the following problem instance $\mathcal{H}$. At the beginning, there are two queries arriving deterministically with a reward 1. Then, over the time interval $[0, 1]$, there are queries with reward $r_1 > 1$ arriving according to a Poisson process with rate λ. At last, there is one query with a reward $\frac{r_2}{\epsilon}$ arriving with a probability ϵ for some small $\epsilon > 0$.

Obviously, since $r_1 > 1$ and ϵ is set to be small, the prophet will first serve the last query as long as it arrives, and then serve the queries with a reward r_1 as much as possible, and at least serve the first two queries. Then, we have that

$$\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})] = \hat{V} := r_2 + 2\cdot\exp(-\lambda) + (r_1+1)\cdot\lambda\cdot\exp(-\lambda) + 2r_1\cdot(1-(\lambda+1)\cdot\exp(-\lambda) + O(\epsilon)$$

Moreover, for any online algorithm π, we consider the following situations separately based on the number of the first two queries that π will serve.

(i). If π will always serve the first two queries, then it is obvious that $\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^\pi(\boldsymbol{I})] = 2$.

(ii). If π serves only one of the first two queries, then the optimal way for π to serve the second query will depend on the value of r_1 and r_2. To be more specific, if $r_1 \geq r_2$, then the optimal way is to serve the query with reward r_1 as long as it arrives, and if $r_1 < r_2$, then the optimal way is to reject all the arriving queries with reward r_1 and only serve the last query. Thus, it holds that

$$\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^\pi(\boldsymbol{I})] \leq V(1) := 1 + \exp(-\lambda)\cdot r_2 + (1-\exp(-\lambda))\cdot\max\{r_1, r_2\} + O(\epsilon)$$

(iii). If π rejects all the first two queries, then conditioning on there are more than one queries with reward r_1 arriving during the interval $[0, 1]$, the optimal way for π is to serve both queries with reward r_1 if $r_1 \geq r_2$ and only serve one query with reward r_1 if $r_1 < r_2$. Then, it holds that

$$\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^\pi(\boldsymbol{I})] \leq V(2) := \exp(-\lambda)\cdot r_2 + \lambda\cdot\exp(-\lambda)\cdot(r_1+r_2) + (1-(\lambda+1)\cdot\exp(-\lambda)\cdot(r_1+\max\{r_1, r_2\})$$

Thus, we conclude that for any online algorithm π, it holds that

$$\frac{\mathbb{E}_{\pi,\boldsymbol{I}\sim\boldsymbol{F}}[V^\pi(\boldsymbol{I})]}{\mathbb{E}_{\boldsymbol{I}\sim\boldsymbol{F}}[V^{\mathrm{off}}(\boldsymbol{I})]} \leq g(r_1, r_2, \lambda) := \frac{\max\{V(1), V(2), 2\}}{\hat{V}}$$

where we can neglect the $O(\epsilon)$ term by letting $\epsilon \to 0$. In this way, we can focus on the following optimization problem

$$\inf_{r_1>1, r_2>1, \lambda} g(r_1, r_2, \lambda)$$

to obtain the upper bound of the guarantee of any online algorithm relative to the prophet's value. We can numerically solve the above problem and show that when $r_1 = r_2 = 1.4119$, $\lambda = 1.2319$, the value of $g(r_1, r_2, \lambda)$ reaches its minimum and equals 0.6269, which completes our proof. ∎