Improved Results for Knapsack with Removal

Matthias Gehnen
Department of Computer Science, RWTH Aachen University, Germany

Kübra Güven
Department of Computer Science, RWTH Aachen University, Germany

Valentin Hächler
Department of Computer Science, ETH Zurich, Switzerland

Dennis Komm
Department of Computer Science, ETH Zurich, Switzerland

Richard Královič
Department of Computer Science, ETH Zurich, Switzerland

Abstract

We study the proportional online knapsack problem with removal. For randomized algorithms, we tighten the gap between the current lower and upper bounds on the expected competitive ratio by presenting a lower bound of roughly 1.27. We further study this problem under the model of online algorithms with predictions. Our lower bound arguments are agnostic to the type of available prediction, which makes them very general. For deterministic algorithms, we provide a tightly matching upper bound on the competitive ratio for a specific kind of weight prediction.

2012 ACM Subject Classification Theory of computation → Adversary models

Keywords and phrases Online computation, competitive analysis, knapsack problem, predictions

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.51

1 Introduction

The knapsack problem is one of Karp's famous 21 NP-complete problems [12] and it is covered in many undergraduate courses on algorithm design. In the problem formulation, we are given a knapsack capacity and a set of items with each item having a weight and a value. The goal is to find a subset of the items that still fits into the knapsack while maximizing the sum of the values of these items. The *simple* or *proportional knapsack problem* simplifies the problem statement by setting each item's value to its weight; i.e., the heavier an item, the more profit it implies, and the objective therefore becomes to maximize the weight of a subset of items, which still meets the capacity requirement. This variation of the problem is closely related to the *subset sum problem*, where we try to find a subset of items which sum up exactly to some given target value.

In the online version of the proportional knapsack problem [13], an *online algorithm* is given a knapsack of some known capacity and then it is fed a sequence of items that arrive one at a time in consecutive time steps. When an item is presented, the algorithm has to decide whether to pack that item or not, without knowing which items might arrive later on. In the classical problem statement, if an item is packed, it cannot be removed later to make space in the knapsack for other items; and if an item is not packed when offered, it cannot be packed at any later point in time. To evaluate the performance of an online algorithm, we compare the profit (gain, utility) of its solution to that of an optimal solution for the given instance; which of course is only computable once all items are revealed.

In this paper, we study a variant of the online proportional knapsack problem *with removal*. This means that one of the restrictions of the classical variant is lifted in that items that have been packed can now be removed from the knapsack at a later time step. We improve the currently best lower bound on randomized algorithms and study both deterministic and randomized algorithms that receive specific kinds of predictions about the yet unrevealed parts of the input.

1.1 Preliminaries

An instance I of an online problem is a sequence of n requests $I = (x_1, \ldots, x_n)$, which are revealed gradually in n consecutive time steps. For the proportional knapsack problem, we assume that the knapsack has a capacity of 1 and thus $0 \leq x_i \leq 1$ for each $i \in \{1, \ldots, n\}$ represents an item of weight x_i. An online algorithm $\textsc{Alg}$ produces the output $\textsc{Alg}(I) = (y_1, \ldots, y_n)$ where y_i only depends on $x_1, \ldots, x_i$; an answer essentially defines whether x_i is packed or not. The *profit of* $\textsc{Alg}$ *on* I is denoted by $\text{profit}(\textsc{Alg}(I))$; similarly, $\text{profit}(\textsc{Opt}(I))$ denotes the *optimal profit for* I. Note that both values are at most 1 for the proportional knapsack problem. In the variant we study in this paper, $\textsc{Alg}$ may also specify a number of already-packed items currently in the knapsack that should be removed in the current time step to make space.

The *competitive ratio* of an online algorithm $\textsc{Alg}$ for such an online maximization problem[1] is defined as

$$c = \sup_{I \in \mathcal{I}} \left(\frac{\text{profit}(\textsc{Opt}(I))}{\text{profit}(\textsc{Alg}(I))} \right) .$$

The reciprocal $f := 1/c$ of the competitive ratio is called the *fill ratio* of the knapsack, which we will use in some parts of this paper to make arguments and calculations more intuitive. A *randomized online algorithm* $\textsc{Rand}$ is allowed to base its answers on the outcome of a random source when computing its solution $\textsc{Rand}(I)$ for a given instance I. With this, $\text{profit}(\textsc{Rand}(I))$ becomes a random variable. The *expected competitive ratio* is defined as

$$c = \sup_{I \in \mathcal{I}} \left(\frac{\text{profit}(\textsc{Opt}(I))}{\mathbb{E}[\text{profit}(\textsc{Rand}(I))]} \right) ,$$

and we define the *expected fill ratio* in the obvious way.

Recently, a field called *online algorithms with predictions* [6] received a lot of attention in the community,[2] which is motivated by advances in machine learning. Here, the algorithm receives information about the yet unrevealed parts of the input, which however is not necessarily correct; but only with some certain probability, say. The goal is to design algorithms that make good use of the predictions and obtain a small competitive ratio if the predictions are correct (*consistency*), and that are not too bad if the predictions are incorrect (*robustness*). The model is related to that of *online algorithms with advice* [4], but differs in some important aspects: as mentioned, the predictions may be faulty and they are usually restricted to encode a specific part of the input. For the knapsack problem, different kinds of predictions have been studied, which we will briefly discuss in the next subsection.

---

1 In the literature an online algorithm is commonly called c-competitive if a constant α exists such that $\text{profit}(\textsc{Opt}(I)) \leq c \cdot \text{profit}(\textsc{Alg}(I)) + \alpha$; if $\alpha = 0$, $\textsc{Alg}$ is called *strictly c-competitive.* Since for the proportional knapsack problem, the profit of any (also the optimal) solution is bounded from above by 1, we can as well restrict ourselves to the strict competitive ratio [13].

2 Currently almost 400 papers are listed at `https://algorithms-with-predictions.github.io`, although not all use a model of online computation as used in this paper.

1.2 Related Work

Online knapsack is by now a well-studied problem, with a rich literature dating back to the mid-90s [15]. Over the last thirty years, many different variants have been introduced and analyzed. In particular, the ability to remove items was added a few years after the initial work, while also other directions of giving the adversary (who creates a worst-case instance for the algorithm) less power were explored [3]. Iwama and Taketomi [11] studied the *proportional knapsack problem with removal* and supplied a sharp bound on the competitive ratio of deterministic online algorithms, namely the golden ratio $(\sqrt{5}+1)/2$. Han et al. [9] provided different hardness results on randomized online algorithms for both the proportional and general knapsack problem, and in particular bounded the expected competitive ratio of any randomized online algorithm for the former from below by 1.25 and from above by $1/0.7 \approx 1.429$.

Böckenhauer et al. studied the proportional knapsack problem both without [5] and with [2] removal with respect to *online algorithms with advice*. One of the most notable results is that, if removal is not allowed, still a single bit of information on the unrevealed parts of the input allows to design a 2-competitive algorithm, while the competitive ratio cannot be bounded at all without this bit. If removal is allowed, a single bit of advice even allows for a competitive ratio of roughly 1.38 [3], which follows from the results of Iwama and Taketomi [11]. As noted, the model of *online algorithms with predictions* was established more recently. Work on the online knapsack problem in this field can be subdivided into different kinds of knapsack variants in combination with different kinds of predictions that were analyzed. Most of the work studies the general version of the problem and assumes some specific bound on the value-weight-ratio of the input items [7, 10].

For the proportional setting (where this ratio is always 1), predictions on the items' weights have been looked into, which is closest to our setting. Gehnen et al. [8] studied a model where an approximation of the whole input is known in advance. In particular, every item's weight is predicted with a multiplicative error bounded by some $\delta \geq 0$. Balabán et al. [1] applied this model to the proportional knapsack problem with removal. They proved a tight bound on the competitive ratio of $(\sqrt{\delta^2+10\delta+9}-\delta+3)/4$ if $0<\delta<1$; note that this bound tends to 1.5 if δ tends to 0. If $\delta \geq 1$, the competitive ratio is bounded by the golden ratio. Balabán et al. also studied the setting where the error is additive instead of multiplicative, in which case the tight bound becomes exactly $(3-2\delta)/(2-2\delta)$ if $0<\delta<1/2$, and again the golden ratio if $\delta \geq 1/2$.

For more on the history of the online knapsack problem and its numerous variants as well as recent results, we refer to a 2026 survey [3].

1.3 Contribution

We study the online proportional knapsack problem with removal under different lenses. In Section 2, we introduce an LP-based lower bound technique that allows us to lift the so far best-known lower bound on the expected competitive ratio of randomized algorithms. Specifically, in Section 3, we present a lower bound of roughly 1.27, improving the aforementioned result of 1.25 by Han et al. [9].

In Section 4, we study the problem with predictions, with a particular focus on the limits of predictions in general. For both deterministic and randomized online algorithms, we present lower bounds that are rather straightforward while not making any assumption on the nature of the predictions. Intriguingly, in Subsection 4.1, we then prove that the deterministic lower bound is tight since there is a powerful kind of weight prediction that allows to match the competitive ratio implied by the lower bound. Conversely, we were not able to also provide such a sharp bound for randomized online algorithms, which we discuss in Subsection 4.2.

2 A Lower Bound Technique Based on Linear Programming

We now focus on a special set of hard instances that will help us to bound the fill ratio (and thus competitive ratio) of any online algorithm for the proportional knapsack with removal. Each of the instances starts with a fixed prefix $(a_1, \ldots, a_n)$ of n items (defined by their weight) with $0 \leq a_i \leq 1$, where no two items of the sequence fit together into the knapsack (i.e., $a_i + a_j > 1$ for all i, j). After presenting this fixed prefix, the instance either ends or a single last item arrives from a fixed set $\{b_1, \ldots, b_n\}$. For simplicity, the first case can be described as presenting an item of value zero as the last item. The resulting set of instances is then $\{I_i\}_{1 \leq i \leq n}$.

After receiving the fixed prefix, any randomized online algorithm Rand may keep at most one item, since no two items of the prefix fit together in the knapsack. Without loss of generality, Rand keeps exactly one item, because removal is always possible anyway. Thus, Rand defines a probability distribution over the items $a_1, \ldots, a_n$, where a_i is kept with probability p_i.

Let $A_{i,j}$ be defined as the best profit that can be obtained when item a_j is kept after reading the fixed prefix and b_i is the last item. In particular, if $a_j + b_i \leq 1$, then $A_{i,j} = a_j + b_i$; otherwise, $A_{i,j} = \max(a_j, b_i)$. The expected profit obtained by Rand on instance I_i can then be at most $\sum_{j=1}^{n} p_j \cdot A_{i,j}$.

Now let $\text{Opt}_i := \text{profit}(\text{Opt}(I_i))$ be the optimal profit for instance I_i. An upper bound on the expected fill ratio (and with this, a lower bound on the expected competitive ratio c) of Rand can be written as

$$\frac{1}{c} \leq \sum_{j=1}^{n} p_j \cdot \frac{A_{i,j}}{\text{Opt}_i}.$$

Consider the matrix $\mathbf{B} = (A_{i,j}/\text{Opt}_i)_{i,j}$ of size $n \times n$ and rewrite the lower bound as

$$\mathbf{B} \cdot \mathbf{p}c \geq \mathbf{1}, \tag{1}$$

where $\mathbf{p} = (p_1, \ldots, p_n)^\top$. A necessary condition for any algorithm to reach competitive ratio c is that there are non-negative $p_1, \ldots, p_n$ that sum to 1 and satisfy (1). If we substitute $\mathbf{x} = \mathbf{p} \cdot c$, it follows that $c = x_1 + \ldots + x_n$, and we can rewrite (1) as

$$\begin{aligned} \mathbf{B} \cdot \mathbf{x} &\geq \mathbf{1}, \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned} \tag{2}$$

Solving the LP defined by (2) with the objective to minimize $\mathbf{1}^\top \mathbf{x}$ yields the best possible competitive ratio c that can be obtained by Rand. As we did not put any restrictions on Rand, this value is a valid lower bound for any randomized online algorithm for the proportional knapsack problem with removal.

In order to improve the lower bound on the competitive ratio, we need to choose the values of $(a_i)_{1 \leq i \leq n}$ and $(b_j)_{1 \leq j \leq n}$ such that the solution of the described LP is as large as possible. The lower bound 1.25 from Han et al. [9] can be improved even for $n = 3$ by choosing $\mathbf{a} = (1 - u, u, v)$ and $\mathbf{b} = (0, u, 1 - u)$ for $u = 0.645$ and $v = 0.697$, yielding a lower bound of a little less than 1.26 on the expected competitive ratio.

3 Improved Lower Bound for Randomized Algorithms

We now use longer instances to improve the lower bound argument from Section 2. Let u and v be fixed constants such that $0.5 \leq u \leq v$. We fix the first item a_1 of the instance prefix $(a_i)_{i=1}^{n+2}$ to be $1-u$, and the remaining items of the sequence are equally spaced between u and v, i.e.,

$$a_1 := 1-u, \quad a_i := u + \frac{i-2}{n}(v-u) \quad \forall i \in \{2, \dots, n+2\} \; .$$

The suffix items are chosen as complements of the first $n-1$ prefix items as

$$b_1 := 0, \quad b_j := 1 - a_{n+3-j} = 1 - v + \frac{j-1}{n}(v-u) \quad \forall j \in \{2, \dots, n+1\}, \quad b_{n+2} := u \; . \tag{3}$$

We use $\textsc{Opt}_i$ and $\mathbf{B}_{i,j}$ as defined in Section 2. Fixing $c_i := u + (i/n)(v-u)$, the values of $\textsc{Opt}$ and $\mathbf{B}$ are then

$$\begin{aligned}
&\textsc{Opt}_1 = v, \quad \textsc{Opt}_i = 1 \quad \forall i \geq 2 \; , \\
&\quad \mathbf{B}_{1,1} = \frac{1-u}{v} \; , \\
&\quad \mathbf{B}_{1,j} = \frac{c_{j-2}}{v} \quad \forall j \in \{2, \dots, n+2\} \; , \\
&\quad \mathbf{B}_{i,1} = 2 - u - v + \frac{i-1}{n}(v-u) \quad \forall i \in \{2, \dots, n+1\} \; , \\
&\quad \mathbf{B}_{i,j} = c_{j-2} + 1 - v + \frac{i-1}{n}(v-u) \quad \forall i \in \{2, \dots, n+1\}, \; j \in \{2, \dots, n+3-i\} \; , \\
&\quad \mathbf{B}_{i,j} = c_{j-2} \quad \forall i \in \{2, \dots, n+1\}, \; j \in \{n+4-i, \dots, n+2\} \; , \\
&\mathbf{B}_{n+2,1} = 1 \; , \\
&\mathbf{B}_{n+2,j} = c_{j-2} \quad \forall j \geq 2 \; .
\end{aligned}$$

Note that the definition of $\mathbf{B}_{i,j}$ for the case $i \in \{2, \dots, n+1\}$ and $j \in \{n+4-i, \dots, n+2\}$ holds since

$$c_{j-2} \geq \frac{1}{2} \geq 1 - v + \frac{i-1}{n}(v-u) \; .$$

The matrix $\mathbf{B}$ can also be described as

$$\mathbf{B} = \begin{pmatrix}
\frac{1-u}{v} & \frac{u}{v} & \dots & \frac{c_{j-2}}{v} & & & \dots & 1 \\
\vdots & & \ddots & & & & & \vdots \\
2-u-v+\frac{i-1}{n}(v-u) & & \dots & c_{j-2}+1-v+\frac{i-1}{n}(v-u) & \dots & c_{j-2} & \dots & \\
\vdots & & \ddots & & & & & \vdots \\
1 & u & \dots & c_{j-2} & & & \dots & v
\end{pmatrix} .$$

As explained in Section 2, the solution of the following LP constitutes a valid lower bound on the expected competitive ratio of any randomized online algorithm:

$$\mathbf{B} \cdot \mathbf{x} \geq \mathbf{1}, \quad \mathbf{x} \geq \mathbf{0}, \quad \min \mathbf{1}^\top \mathbf{x} \tag{4}$$

But then of course any feasible solution of the dual LP

$$\mathbf{B}^\top \cdot \mathbf{y} \leq \mathbf{1}, \quad \mathbf{y} \geq \mathbf{0}, \quad \max \mathbf{1}^\top \mathbf{y} \tag{5}$$

is also a valid lower bound.

Now let us fix

$$u \coloneqq \frac{\sqrt{5}-1}{2} \approx 0.61803 ,$$

then take v to satisfy $(1-u)/(1-v) = \mathrm{e}^{1/u+v-2}$, which implies

$$v \approx 0.73042 ,$$

and finally set

$$p \coloneqq \frac{1-u}{u} + \frac{2u-1}{uv} + \frac{1-v}{v} \ln\left(\frac{1-u}{1-v}\right) \approx 1.26958 .$$

The following theorem shows that p is a lower bound on the expected competitive ratio of any randomized online algorithm. The main idea of the proof is to consider a certain fixed assignment of $\mathbf{y}$, show that such an assignment is a valid solution of the dual LP, and that the value of this solution converges to p when n tends to infinity.

▶ **Theorem 1.** *Let q and r be defined as*

$$q \coloneqq \frac{1}{u} - 1 - \ln\left(\frac{1-u}{1-v}\right) \approx 0.26989 \quad \textit{and} \quad r \coloneqq \frac{2u-1}{uv} \approx 0.52294 .$$

The following vector $\mathbf{y}$ is a valid solution to the dual LP (5)*:*

$$y_1 \coloneqq q, \quad y_{n+2} \coloneqq r, \quad y_i \coloneqq \frac{v-u}{vn} \cdot \frac{1}{1-v+\frac{i-1}{n}(v-u)} \quad \forall i \in \{2,\dots,n+1\}$$

The value of the solution converges to $p \approx 1.26958$ with n tending to infinity.

Proof. The inequalities of the dual LP are as follows:

For the first row of $\mathbf{B}^\top$, we have

$$y_1 \frac{1-u}{v} + \sum_{i=2}^{n+1} y_i \Big(2-u-v+\frac{i-1}{n}(v-u)\Big) + y_{n+2} \leq 1 .$$

For row $j+2$ of $\mathbf{B}^\top$ with $0 \leq j \leq n$, we have

$$y_1 \frac{c_j}{v} + \sum_{i=2}^{n+1} y_i \cdot c_j + \sum_{i=2}^{n+1-j} y_i \Big(1-v+\frac{i-1}{n}(v-u)\Big) + y_{n+2} \cdot c_j \leq 1 . \tag{6}$$

We can interpret y_2 to y_{n+1} as rectangles of width $(v-u)/n$ and height $1/v \cdot (1-v+((i-1)/n)(v-u))^{-1}$, and we can therefore bound their sums by

$$\begin{aligned}
&\sum_{i=2}^{n+1} y_i \leq \int_{1-v}^{1-u} \frac{1}{v} \cdot \frac{1}{x} \,\mathrm{d}x ,\\
&\sum_{i=2}^{n+1} y_i (1-v+\frac{i-1}{n}(v-u)) \leq \int_{1-v}^{1-u} \frac{1}{v} \,\mathrm{d}x ,\\
&\sum_{i=2}^{n+1-j} y_i \leq \int_{1-v}^{1-c} \frac{1}{v} \cdot \frac{1}{x} \,\mathrm{d}x, \quad c = u + \frac{j}{n}(v-u) ,\\
&\sum_{i=2}^{n+1-j} y_i (1-v+\frac{i-1}{n}(v-u)) \leq \int_{1-v}^{1-c} \frac{1}{v} \,\mathrm{d}x, \quad c = u + \frac{j}{n}(v-u) .
\end{aligned} \tag{7}$$

If we apply the bounds (7) to the inequalities (6), we get

$$
\begin{aligned}
&q\frac{1-u}{v} + \int_{1-v}^{1-u} \frac{1}{v}\cdot\frac{1}{x}(1-u+x)\,\mathrm{d}x + r \le 1\,,\\
&q\frac{c_j}{v} + \int_{1-v}^{1-u} \frac{1}{v}\cdot\frac{c_j}{x}\,\mathrm{d}x + \int_{1-v}^{1-c_j} \frac{1}{v}\,\mathrm{d}x + rc_j \le 1, \quad u \le c_j \le v\,,
\end{aligned}
\tag{8}
$$

and this new set of inequalities is a sufficient condition to guarantee that the inequalities (6) are true for every n. After evaluating the integrals, the inequalities (8) are equivalent to

$$
\begin{aligned}
&q\frac{1-u}{v} + \frac{1-u}{v}\ln\frac{1-u}{1-v} + \frac{v-u}{v} + r \le 1\,,\\
&\frac{c_j}{v}\left(q + \ln\frac{1-u}{1-v} + rv\right) + \frac{1}{v}\ln\frac{1-c_j}{1-v} \le 1\,.
\end{aligned}
\tag{9}
$$

It is straightforward to verify that, after substituting the values of r and q, the inequalities (9) are satisfied for all values c_j between u and v.

Furthermore, we can also bound the output value of the LP in a similar manner by

$$
\begin{aligned}
\mathbf{1}_{n+2}^{\top}\mathbf{y} = \sum_{i=1}^{n+2} y_i &= q + r + \sum_{i=2}^{n+1}\frac{1}{v}\cdot\frac{v-u}{n}\,\frac{1}{1-v+\frac{i-1}{n}(v-u)}\\
&= q + r + \sum_{i=1}^{n+1}\frac{1}{v}\cdot\frac{v-u}{n}\,\frac{1}{1-v+\frac{i-1}{n}(v-u)} - \frac{v-u}{vn}\,\frac{1}{1-v}\\
&\ge q + r + \int_u^v \frac{1}{v}\cdot\frac{1}{1-x}\,\mathrm{d}x - \frac{v-u}{vn}\,\frac{1}{1-v}\\
&= q + r + \frac{1}{v}\cdot\ln\left(\frac{1-u}{1-v}\right) - \frac{v-u}{vn}\,\frac{1}{1-v}\,,
\end{aligned}
$$

where the last term converges to 0 if n tends to infinity, which proves the claim. ◀

Using a similar reasoning, we can also prove that the value of p is the best possible bound implied by the LP (4). In particular, we show that there is a feasible solution of the primal LP that converges to p when n tends to infinity. This claim of optimality does not rule out a different choice of the prefix sequence (a_i) and the last item set $\{b_j\}$. The current choice, however, contains all values of a_i from interval (u, v) if n tends to infinity, and the values of u and v were chosen such that the bound is the strongest possible. Furthermore, if we restrict the lower bound analysis to a "single round," where the algorithm sees the whole prefix, can decide on a probability distribution over the items kept, and then the remaining set of items arrives, the prefix sequence with a single item below $1/2$ is the hardest case: For any other sequence, the algorithm could consider all subsets of items that fit into the knapsack and cannot be extended by another item, and thus reduce the input to the case of a single item smaller than $1/2$. Thus, our results provide solid evidence that stronger lower bounds are not possible with a single-round analysis.

▶ **Theorem 2.** *Let q, r, and s be defined as*

$$
q := \frac{1-v}{u}, \quad r := \frac{(1-v)(2u-1)}{u(1-u)}, \quad \textit{and} \quad s := \frac{(v-1)(1-2u+u\ln(\frac{v-1}{u-1}))}{uv}\,.
$$

The following vector $\mathbf{x}$ *is a valid solution to the primal LP* (4)*:*

$$x_1 \coloneqq q, \quad x_2 \coloneqq r, \quad x_{n+2} \coloneqq s + \frac{v-u}{n(1-v)},$$
$$x_i \coloneqq (1-v)\frac{v-u}{n(1-u-\frac{i-2}{n}(v-u))^2} \quad \forall i \in \{3,\dots,n+1\}$$

The value of the solution converges to p *with* n *tending to infinity.*

Proof. The constraints of the primal LP are as follows:

For the first row of $\mathbf{B}$, we have

$$x_1\frac{1-u}{v} + x_2\frac{u}{v} + \sum_{i=3}^{n+1} x_i \frac{u+\frac{i-2}{n}(v-u)}{v} + x_{n+2} \geq 1\ .$$

For row $j+1$ of $\mathbf{B}$, where $1 \leq j \leq n, d_j \coloneqq 1-v+\dfrac{j}{n}(v-u)$, we have

$$x_1(1-u+d_j) + x_2(u+d_j) + \sum_{i=3}^{n+1} x_i\Big(u + \frac{i-2}{n}(v-u)\Big) + \sum_{i=3}^{n+2-j} x_i \cdot d_j + x_{n+2}\cdot v \geq 1\ .$$

For row $n+2$ of $\mathbf{B}$, we have

$$x_1 + x_2 \cdot u + \sum_{i=3}^{n+1} x_i\Big(u+\frac{i-2}{n}(v-u)\Big) + x_{n+2}\cdot v \geq 1\ .$$

We can bound the constraints by integrals similarly to the proof of Theorem 1, and reformulate the sufficient condition for the inequalities of the LP to

$$q\frac{1-u}{v} + r\frac{u}{v} + s + \int_u^v \frac{1-v}{v}\frac{y}{(1-y)^2}\,\mathrm{d}y \geq 1\ ,$$
$$q(1-u+d_j) + r(u+d_j) + sv + \int_u^v (1-v)\frac{y}{(1-y)^2}\,\mathrm{d}y + \int_u^{1-d_j}(1-v)\frac{d_j}{(1-y)^2}\,\mathrm{d}y \geq 1\ ,$$
$$q + ru + sv + \int_u^v (1-v)\frac{y}{(1-y)^2}\,\mathrm{d}y \geq 1\ .$$

Since $1-v \leq d_j \leq 1-u$, it is sufficient to calculate the integrals and substitute the chosen values of q, r, and s to verify that the inequalities hold, implying that the chosen solution of $\mathbf{x}$ is a valid solution of the linear program for any value of n.

We can also bound the output value of the LP by

$$\mathbf{1}_{n+2}^\top\mathbf{x} = \sum_{i=1}^{n+2} x_i \leq q + r + s + (1-v)\int_u^v \frac{1}{(1-y)^2}\,\mathrm{d}y + \frac{v-u}{n(1-v)}\ .$$

The last term vanishes in the limit, leaving us with

$$\lim_{n\to\infty} \mathbf{1}_{n+2}^\top\mathbf{x} = q + r + s + (1-v)\int_u^v \frac{1}{(1-y)^2}\,\mathrm{d}y\ .$$

Calculating the integral and substituting q, r, and s concludes the proof. ◀

4 Online Algorithms with Predictions

We now assume that an online algorithm may receive information, so-called *predictions*, about the unknown parts of the input. As described in Subsection 1.2, there are different kinds of predictions for the knapsack problem; for the proportional knapsack problem with removal, recent work essentially assumes that the items' weights are predicted within certain error bounds. Indeed, predictions are not fully reliable, and we are particularly interested in the two cases that all predictions are correct or that there is some (even a possibly very small) error in some of the predictions.

For the lower bounds presented in this section, we do not make any assumption on the nature of the predictions, which makes our corresponding statements particularly strong. We then consider specific powerful predictions to put these results into perspective.

4.1 Deterministic Algorithms

We present a tight bound on the competitive ratio of deterministic online algorithms with predictions. As noted in Subsection 1.2, without predictions, there is a $(\sqrt{5}+1)/2$-competitive deterministic algorithm and this bound is tight [11]. It is consequently sufficient to focus on algorithms that have a competitive ratio better than this threshold in the case of correct predictions.

At first, we present a lower bound on the competitive ratio of deterministic online algorithms that is valid for any kind of prediction.

▶ **Theorem 3.** *Let* Alg *be any deterministic algorithm for proportional knapsack with removal that is c-competitive for correct predictions, where $1 \le c \le (\sqrt{5}+1)/2$. Then* Alg *is no better than $1/(c-1)$-competitive for incorrect predictions.*

Proof. Let $\varepsilon > 0$ and consider the behavior of Alg on the input $(1-1/c+2\varepsilon, 1/c-\varepsilon, 1/c-2\varepsilon)$ with the corresponding correct prediction. After receiving the first two items, Alg can keep only one of them.

If Alg keeps the first one and no third item arrives, i.e., the prediction is incorrect, the competitive ratio of Alg is bounded by $(1-\varepsilon c)/(c-1+2\varepsilon c)$; thus, the claim of the theorem holds since ε can be arbitrarily small. If Alg keeps the second item and the prediction is correct, the competitive ratio is $c/(1-\varepsilon c)$, which contradicts the assumption. ◀

The presented lower bound is tight, since for sufficiently powerful predictions, there is an algorithm that achieves the corresponding competitive ratios. In particular, we assume that the predictions encode properties about an optimal solution of the given instance. Such models have, e.g., been studied by Lechowicz et al. [14], who assumed specific *frequency predictions* or Daneshvaramoli et al. [7], who investigated *succinct predictions*.

▶ **Theorem 4.** *For any c with $1 \le c \le (\sqrt{5}+1)/2$, there is a deterministic algorithm* Alg *for the knapsack problem with removal that is c-competitive for correct predictions and $(1/(c-1))$-competitive for incorrect predictions.*

Proof. Recall that $f := 1/c$ denotes the fill ratio, and let I be the analyzed instance. We classify items of I smaller than $1-f$ as *small*, items of weight at least $1-f$ and smaller than $(1-f)/f$ as *medium*, items of weight at least $(1-f)/f$ and smaller than f as *large*, and items of weight at least f as *huge*. Let I' be an instance that is identical to I, but with all small and huge items removed. The prediction contains the set of large items that are included in an optimal solution of I'.

Alg works as follows. Whenever a huge item is encountered, it is kept as the only item and Alg terminates. Otherwise, the input item is kept. If the total weight of kept items exceeds the knapsack capacity 1, a subset of items is chosen by greedily selecting items according to the following priorities: at first, large items predicted to be in the optimal solution are selected in an arbitrary order; then, all medium and remaining large items are chosen from the smallest to the largest; at the end, small items are selected in an arbitrary order. If any small item was removed, ALG terminates.

If I contains any huge item, ALG computes a solution with profit at least f, thus a competitive ratio of at most $1/f = c$ is guaranteed. For the remaining analysis, assume that there are no huge items.

If, at any time, ALG removes any small item, it provides a solution with a profit of at least $1 - (1 - f)$, which guarantees a competitive ratio of at most c. Thus, we may assume that ALG keeps all small items of I. Let s be the total weight of all small items. Since the small items have the smallest priority, ALG keeps the same medium and large items of I as it would keep when working on I', and it follows that $\text{profit}(\text{ALG}(I)) = s + \text{profit}(\text{ALG}(I'))$. As $\text{profit}(\text{OPT}(I)) \leq s + \text{profit}(\text{OPT}(I'))$, the competitive ratio of ALG on I can be bounded by

$$\frac{\text{profit}(\text{OPT}(I))}{\text{profit}(\text{ALG}(I))} \leq \frac{s + \text{profit}(\text{OPT}(I'))}{s + \text{profit}(\text{ALG}(I'))} \leq \frac{\text{profit}(\text{OPT}(I'))}{\text{profit}(\text{ALG}(I'))} .$$

Hence, it is sufficient to consider instances that contain only medium and large items.

Let us first consider the case of an incorrect prediction. If there is any large item predicted to be in the optimal solution, ALG takes at least one such item. In that case, $\text{profit}(\text{ALG}(I)) \geq (1-f)/f = c-1$, which guarantees a competitive ratio of at most $1/(c-1)$. Conversely, assume there is no such item. Then, at every step, the i-th smallest item kept by ALG is at most as large as the i-th smallest item kept by OPT. Therefore, ALG holds at least as many items as OPT at any time step. In the worst case, OPT packs an item of weight $1-f$ instead of one of weight f, resulting in a competitive ratio of at most $f/(1-f) = 1/(c-1)$.

The last case to consider is the behavior of ALG in the presence of a correct prediction. In that case, ALG keeps the correct set of large items and the only difference to the optimal solution are the medium items. Similarly to the previous case, ALG always keeps at least as many items in its knapsack as OPT. In the worst case, ALG keeps a medium item of weight $1 - f$ instead of one of weight $(1 - f)/f$. This results in a competitive ratio of at most

$$\frac{\frac{1-f}{f}}{1-f} = \frac{1}{f} = c ,$$

which proves the claim. ◀

The bound of Theorems 3 and 4 is shown in Figure 1.

4.2 Randomized Algorithms

We now study the power of randomized online algorithms in the predictions model. We again start with a general lower bound that makes no assumptions on the predictions' nature.

▶ **Theorem 5.** *Let* RAND *be any randomized algorithm for proportional knapsack with removal that has an expected competitive ratio strictly smaller than c for correct predictions, where* $1 \leq c \leq 1.25$. *Then the expected competitive ratio of* RAND *for incorrect predictions is no better than*

$$\frac{c}{2\sqrt{c-1}} .$$

Proof. Suppose some randomized online algorithm RAND violates the claim, and again consider the expected fill ratio $f := 1/c$. Since RAND is strictly better than c-competitive for correct predictions, it obtains a competitive ratio of $c/(1 + \varepsilon \cdot c) = 1/(f + \varepsilon)$ for some sufficiently small $\varepsilon > 0$.

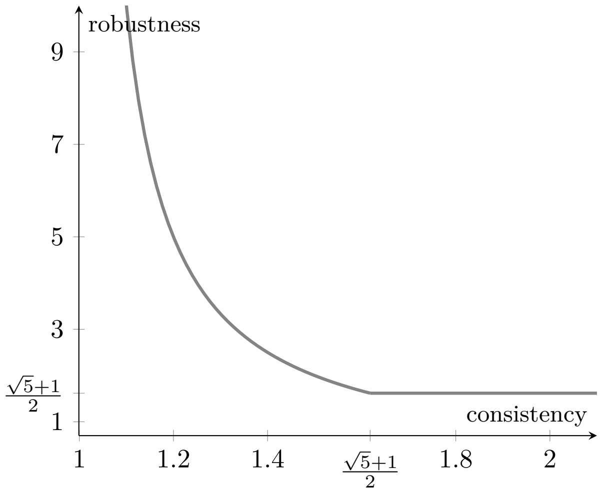

Figure 1 The tight bound on the competitive ratio of deterministic online algorithms; the trade-off for correct (consistency) vs. incorrect predictions (robustness).

Now consider an instance $I \coloneqq (x, 1-x+\varepsilon/2, 1-x)$, where $x \in (\varepsilon/2, 1/2)$; the value of x is chosen later to get the tightest possible bound.

After receiving the second item, Rand may keep only one of the items. Let p be the probability that the first item is kept in the presence of the correct prediction. The following bounds then hold for the competitive ratio of Rand:

$$\begin{aligned}\frac{1}{f+\varepsilon} &\geq \frac{1}{p+(1-p)\left(1-x+\frac{\varepsilon}{2}\right)}\,,\\ f+\varepsilon &\leq 1-x+\frac{\varepsilon}{2}+px-\frac{p\varepsilon}{2} < 1-x+\frac{\varepsilon}{2}+px\,,\\ p &> \frac{f-1+x}{x}\end{aligned}$$

Now consider the instance $I' \coloneqq (x, 1-x+\varepsilon/2)$, which is identical to I on the first two items, but there is no third item. Assume that Rand is provided with incorrect predictions that are identical to the correct predictions for I. Thus, Rand cannot distinguish between these cases after receiving the first two items and consequently keeps the first one with probability p. As a result, we have the following bound on the expected competitive ratio c' of Rand with incorrect predictions:

$$c' \geq \frac{1-x+\frac{\varepsilon}{2}}{p\left(x-1+x-\frac{\varepsilon}{2}\right)+1-x+\frac{\varepsilon}{2}} \geq \frac{1}{1+\frac{2x-1-\frac{\varepsilon}{2}}{1-x+\frac{\varepsilon}{2}}\cdot\frac{f-1+x}{x}} \geq \frac{1}{\frac{f}{1-x}+\frac{1-f}{x}-1} \tag{10}$$

It is straightforward to verify that this lower bound on c' is maximized if

$$x = \frac{1}{1+\sqrt{\frac{f}{1-f}}} = \frac{1}{1+\frac{1}{\sqrt{c-1}}}\,,$$

and plugging this value into (10) yields $c' \geq c/(2\sqrt{c-1})$, proving the claim. ◀

In contrast to the results in Subsection 4.1, we were not able to come up with a matching upper bound. To demonstrate the headroom, we present the following observation about an upper bound for arbitrarily powerful predictions, which however results in a competitive ratio that is quite different from the lower bound of Theorem 5.

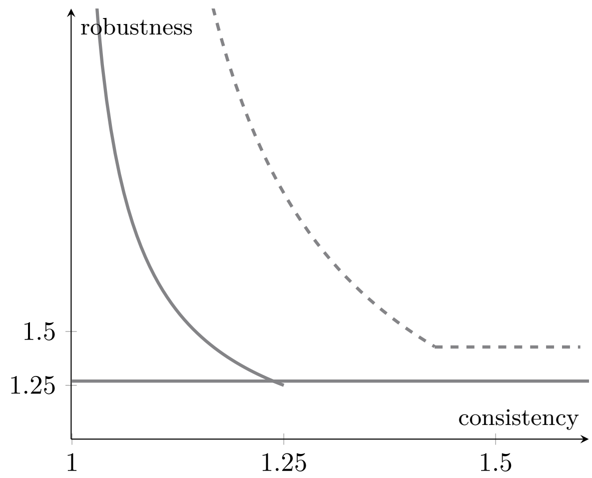

Figure 2 The expected competitive ratio of randomized online algorithms for the proportional knapsack problem with removal—the trade-off for correct (consistency) vs. incorrect predictions (robustness); the upper bound is drawn as a dashed line, the lower bound as a solid line.

▶ **Theorem 6.** *For any p with $0 \leq p < 1$, there is a randomized online algorithm* Rand *for proportional knapsack with removal with competitive ratio $10/(7+3p)$ for correct predictions and $10/(7-7p)$ for incorrect predictions.*

Proof. Assume that the prediction encodes an optimal solution for the current instance I. With probability p, Rand trusts the prediction. With probability $1-p$, Rand ignores the prediction and simulates a $1/0.7$-competitive algorithm for proportional knapsack with removal [9]. If the prediction is correct, the expected profit of Rand is

$$p \cdot \text{profit}(\textsc{Opt}(I)) + (1-p) \cdot 0.7 \cdot \text{profit}(\textsc{Opt}(I)) = (0.7 + 0.3p) \cdot \text{profit}(\textsc{Opt}(I)) \ .$$

If the prediction is incorrect and at the same time trusted, we have no guarantee about the profit of Rand. However, since Rand does not trust the prediction with probability $1-p$, we are in fact guaranteed an expected profit of at least

$$(1-p) \cdot 0.7 \cdot \text{profit}(\textsc{Opt}(I)) = (0.7 - 0.7p) \cdot \text{profit}(\textsc{Opt}(I)) \ ,$$

which proves the claim. ◀

The upper and lower bounds for randomized algorithms are shown in Figure 2. The upper bound is implied by Theorem 6 and the $(1/0.7)$-competitive randomized algorithm of Han et al. [9]. The lower bound is implied by Section 3 and Theorem 5. Note that for values of consistency slightly smaller than 1.25, the robustness lower bound of 1.26958 from Section 3 is already stronger than the bound provided by Theorem 5. This suggests that the result of Theorem 5 could be improved by similar techniques.

5 Conclusion and Future Work

We studied the proportional online knapsack problem under three different lenses: deterministic computation, randomized computation, and computation with predictions. We were able to lift the current lower bound on the expected competitive ratio of randomized online algorithms from 1.25 to roughly 1.27 by introducing an LP-based lower bound technique. We are positive that this technique may have impact beyond this rather specific use case due to its general nature. This remains to be explored in future work.

Besides that, also some more specific questions remain open. Of course, it is still not clear where the actual expected competitive ratio for randomized online algorithms lies. There remains a large gap between our lower bound of roughly 1.27 and the current upper bound of $1/0.7 \approx 1.43$ [9].

Also, a lot remains to be explored in the realm of predictions. We for instance so far lack a clear picture for randomized algorithms with respect to the bounds presented in Subsection 4.2. This is not surprising, as a tight bound in the setting with predictions would imply a tight bound in the setting without predictions.

Then again, while our lower bounds are very robust due to their general nature, our upper bounds use very powerful predictions that encode information about an optimal solution, not simply on the instance. Future work should look into less powerful – and possibly more realistic – predictions.

References

1. Jakub Balabán, Matthias Gehnen, Henri Lotze, Finn Seesemann, and Moritz Stocker. Online knapsack problems with estimates. In *Proceedings of the 50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.MFCS.2025.12.
2. Hans-Joachim Böckenhauer, Fabian Frei, and Peter Rossmanith. Removable online knapsack and advice. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.STACS.2024.18.
3. Hans-Joachim Böckenhauer, Juraj Hromkovic, Dennis Komm, Peter Rossmanith, and Moritz Stocker. A survey of online knapsack problems. *Discrete Applied Mathematics*, 378:492–507, 2026. doi:10.1016/J.DAM.2025.08.011.
4. Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královič, Richard Královič, and Tobias Mömke. Online algorithms with advice: The tape model. *Information and Computation*, 254:59–83, 2017. doi:10.1016/j.ic.2017.03.001.
5. Hans-Joachim Böckenhauer, Dennis Komm, Richard Královič, and Peter Rossmanith. The online knapsack problem: Advice and randomization. *Theoretical Computer Science*, 527:61–72, 2014. doi:10.1016/J.TCS.2014.01.027.
6. Joan Boyar. Online Algorithms with Predictions. In *Proceedings of the 48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272, pages 2:1–2:2, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.2.
7. Mohammadreza Daneshvaramoli, Helia Karisani, Adam Lechowicz, Bo Sun, Cameron Musco, and Mohammad Hajiesmaili. Competitive algorithms for online knapsack with succinct predictions. *CoRR*, abs/2406.18752, 2024. doi:10.48550/arXiv.2406.18752.
8. Matthias Gehnen, Henri Lotze, and Peter Rossmanith. Online simple knapsack with bounded predictions. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.STACS.2024.37.
9. Xin Han, Yasushi Kawase, and Kazuhisa Makino. Randomized algorithms for online knapsack problems. *Theoretical Computer Science*, 562:395–405, 2015. doi:10.1016/J.TCS.2014.10.017.
10. Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. In *34th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS 2021)*, pages 2733–2743, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/161c5c5ad51fcc884157890511b3c8b0-Abstract.html.

11 Kazuo Iwama and Shiro Taketomi. Removable online knapsack problems. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP 2002)*, pages 293–305. Springer, 2002. doi:10.1007/3-540-45465-9_26.

12 Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations, Plenum Press, New York*, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.

13 Dennis Komm. *An Introduction to Online Computation: Determinism, Randomization, Advice*. Springer, 2016. doi:10.1007/978-3-319-42749-2.

14 Adam Lechowicz, Rik Sengupta, Bo Sun, Shahin Kamali, and Mohammad Hajiesmaili. Time fairness in online knapsack problems. In *12th International Conference on Learning Representations (ICLR 2024)*, 2024. URL: https://openreview.net/forum?id=9kG7TwgLYu.

15 Alberto Marchetti-Spaccamela and Carlo Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68:73–104, 1995. doi:10.1007/BF01585758.