# Online Knapsack Problems with a Resource Buffer


**Xin Han**
Dalian University of Technology, Dalian, China
hanxin@dlut.edu.cn

**Yasushi Kawase**
Tokyo Institute of Technology, Tokyo, Japan
kawase.y.ab@m.titech.ac.jp

**Kazuhisa Makino**
Kyoto University, Kyoto, Japan
makino@kurims.kyoto-u.ac.jp

**Haruki Yokomaku**
NTT DATA Mathematical Systems, Tokyo, Japan
dsm4up2c@gmail.com



## Abstract

In this paper, we introduce online knapsack problems with a resource buffer. In the problems, we are given a knapsack with capacity 1, a buffer with capacity $R \geq 1$, and items that arrive one by one. Each arriving item has to be taken into the buffer or discarded on its arrival irrevocably. When every item has arrived, we transfer a subset of items in the current buffer into the knapsack. Our goal is to maximize the total value of the items in the knapsack. We consider four variants depending on whether items in the buffer are removable (i.e., we can remove items in the buffer) or non-removable, and proportional (i.e., the value of each item is proportional to its size) or general. For the general&non-removable case, we observe that no constant competitive algorithm exists for any $R \geq 1$. For the proportional&non-removable case, we show that a simple greedy algorithm is optimal for every $R \geq 1$. For the general&removable and the proportional&removable cases, we present optimal algorithms for small $R$ and give asymptotically nearly optimal algorithms for general $R$.



**2012 ACM Subject Classification** Theory of computation → Online algorithms; Theory of computation → Discrete optimization

**Keywords and phrases** Online knapsack problem, Resource augmentation, Competitive analysis

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2019.28

**Related Version** A full version of the paper is available at [8], `https://arxiv.org/abs/1909.10016`.

**Funding** *Xin Han*: supported by RGC (HKU716412E) and NSFC (11571060).
*Yasushi Kawase*: supported by JSPS KAKENHI Grant Number 16K16005.
*Kazuhisa Makino*: supported by JSPS KAKENHI Grant Number JP24106002, JP25280004, JP26280001, and JST CREST Grant Number JPMJCR1402.


## 1 Introduction

Online knapsack problem is one of the most fundamental problems in online optimization [16, 18]. In the problem, we are given a knapsack with a fixed capacity, and items with sizes and values, which arrive one by one. Upon arrival, we must decide whether to accept the arrived item into the knapsack, and this decision is irrevocable.

In this paper, we introduce a variant of the online knapsack problem, which we call *online knapsack problems with a resource buffer*. Suppose that we have a buffer with fixed capacity in addition to a knapsack with fixed capacity, and items arrive online. Throughout this paper, we assume that the knapsack capacity is 1, and the buffer capacity is $R$ ($\geq 1$). In addition, assume that each item $e$ has a size $s(e)$ and a value $v(e)$. When an item $e$ has arrived, we must decide whether to take it into the buffer or not. The total size of the selected items must not exceed the capacity of the buffer $R$. Further, we cannot change the decisions that we made past, i.e., once an item is rejected, it will never be put into the buffer. We consider two settings: (i) *non-removable*, i.e., we cannot discard items in the buffer, and (ii) *removable*, i.e., we can discard some items in the buffer, and once an item is discarded, it will never be put into the buffer again. After the end of the item sequence, we transfer a subset of items from the buffer into the knapsack. Our goal is to maximize the total value of the items in the knapsack under the capacity constraint. It is worth mentioning that, if $R = 1$, our problem is equivalent to the standard online knapsack problem.

Our model can be regarded as a "partial" resource augmentation model. That is, in the resource augmentation model, the online algorithm can use the buffer for the final result. On the other hand, in our model, the online algorithm uses the buffer only to temporary store items, and it must use the knapsack to output the final result. Moreover, our model can be viewed as a streaming setting: we process items in a streaming fashion, and we can keep only a small portion of the items in memory at any point.

To make things more clear, let us see an example of the online knapsack problem with a resource buffer. Let $R = 1.5$. Suppose that three items $e_1, e_2, e_3$ with $(s(e_1), v(e_1)) = (0.9, 4)$, $(s(e_2), v(e_2)) = (0.7, 3)$, $(s(e_3), v(e_3)) = (0.2, 2)$ are given in this order, but we do not know the items in advance. When $e_1$ has arrived, suppose that we take it into the buffer. Then, for the non-removable case, we need to reject $e_2$ because we cannot put it together with $e_1$. In contrast, for the removable case, we have another option – take $e_2$ into the buffer by removing $e_1$. If $\{e_1, e_3\}$ is selected in the buffer at the end, the resulting value is 4 by transferring $\{e_1\}$ to the knapsack. Note that, in the resource augmentation model, we can obtain a solution with value 6 by selecting $\{e_1, e_3\}$.

## Related work

For the non-removable online knapsack problem (i.e., non-removable case with $R = 1$), Marchetti-Spaccamela and Vercellis [19] showed that no constant competitive algorithm exists. Iwama and Taketomi [9] showed that there is no constant competitive algorithm even for the proportional case (i.e., the value of each item is proportional to its size). The problem has also studied under some restrictions on the input [1, 4, 17, 20].

The removable variant of the online knapsack problem (i.e., removable case with $R = 1$) is introduced by Iwama and Taketomi [9]. They proved that no constant competitive deterministic algorithm exists in general, but presented an optimal $(1 + \sqrt{5})/2$-competitive algorithm for the proportional case. The competitive ratios can be improved by using randomization [5, 7]. In addition, the problem with removal cost has been studied under the name of the *buyback problem* [2, 3, 6, 11, 12].

An online knapsack problem with resource augmentation is studied by Iwama and Zhang [10]. In their setting, an online algorithm is allowed to use a knapsack with capacity $R \geq 1$, while the offline algorithm has a knapsack with capacity 1. They developed optimal $\max\{1, 1/(R-1)\}$-competitive algorithms for the general&removable and proportional&non-removable cases and an optimal $\max\{1, \min\{\frac{1+\sqrt{4R+1}}{2R}, \frac{2}{2R-1}\}\}$-competitive algorithm for the proportional&removable case. All of their algorithms are based on simple greedy strategies. The competitive ratios except for the general&non-removable cases become exactly 1 when $R$ is a sufficiently large real.

In addition, there exist several papers that apply online algorithms to approximately solve the constrained stable matching problems [13–15].

## Our results

We consider four variants depending on whether removable or non-removable, and proportional or general. In this paper, we focus on deterministic algorithms. Our results are summarized in Table 1. To compare our model to the resource augmentation model, we list the competitive ratio for both models in the table. It should be noted that each competitive ratio in our model is at least the corresponding one in the resource augmentation model. Hence, lower bounds for the resource augmentation model are also valid to our model.

For the general&non-removable case, we show that there is no constant competitive algorithm. For the proportional&non-removable case, we show that a simple greedy is optimal and its competitive ratio is $\max\{2, 1/(R-1)\}$. Interestingly, the competitive ratio is equal to the ratio in resource augmentation model for $1 < R \leq 3/2$. For the general&removable case, we present an optimal algorithm for $1 < R \leq 2$. Furthermore, for large $R$, we provide an algorithm that is optimal up to a logarithmic factor. The algorithm partitions the input items into groups according to sizes and values, and it applies a greedy strategy for each group that meets a dynamically adjusted threshold. We will see that the competitive ratio is larger than 1 for any $R$ but it converges to 1 as $R$ goes to infinity. For the proportional&removable case, we develop optimal algorithms for $1 \leq R \leq 3/2$. The basic idea of the algorithms is similar to that of the algorithm for $R = 1$ given by Iwama and Taketomi [9]. Our algorithms classify the items into three types – *small*, *medium*, and *large* – and the algorithms carefully treat medium items. We observe that, as $R$ becomes large, we need to handle more patterns to obtain an optimal algorithm. In addition, for large $R$, we show that the algorithm for the general&removable case is also optimal up to a logarithmic factor.

**Table 1** Summary of the competitive ratios for our model and the resource augmentation model.

| variants | | Our model: $R$ | Our model: lower bound | Our model: upper bound | Resource augmentation: $R$ | Resource augmentation: lower bound | Resource augmentation: upper bound |
|---|---|---|---|---|---|---|---|
| non-removable | prop. | $1$ | $\infty$ [19] | – | $1$ | $\infty$ [19] | – |
| | | $(1,\ \frac{3}{2}]$ | $\frac{1}{R-1}$ [10] | $\frac{1}{R-1}$ (Thm. 4) | $(1,\ 2]$ | $\frac{1}{R-1}$ [10] | $\frac{1}{R-1}$ [10] |
| | | $[\frac{3}{2},\ \infty)$ | 2 (Thm. 3) | 2 (Cor. 5) | $[2,\ \infty)$ | 1 | 1 [10] |
| | gen. | $[1,\ \infty)$ | $\infty$ [19] | – | $[1,\ \infty)$ | $\infty$ [19] | – |
| removable | prop. | $1$ | $\frac{1+\sqrt{5}}{2}$ [9] | $\frac{1+\sqrt{5}}{2}$ [9] | $1$ | $\frac{1+\sqrt{5}}{2}$ [9] | $\frac{1+\sqrt{5}}{2}$ [9] |
| | | $[1,\ \frac{1+\sqrt{2}}{2}]$ | $\frac{1+\sqrt{4R+1}}{2R}$ (Thm. 13) | $\frac{1+\sqrt{4R+1}}{2R}$ (Thms. 14, 15) | $[1,\ \frac{1+\sqrt{2}}{2}]$ | $\frac{1+\sqrt{4R+1}}{2R}$ [10] | $\frac{1+\sqrt{4R+1}}{2R}$ [10] |
| | | $[\frac{1+\sqrt{2}}{2},\ 2-\frac{\sqrt{2}}{2}]$ | $\sqrt{2}$ †) | $\sqrt{2}$ (Thm. 15) | | | |
| | | $[2-\frac{\sqrt{2}}{2},\ 17-9\sqrt{3}]$ | $\frac{\sqrt{16R+1}-1}{2R}$ †) | $\frac{\sqrt{16R+1}-1}{2R}$ †) | $[\frac{1+\sqrt{2}}{2},\ \frac{3}{2}]$ | $\frac{2}{2R-1}$ [10] | $\frac{2}{2R-1}$ [10] |
| | | $[17-9\sqrt{3},\ 2\sqrt{3}-2]$ | $\frac{1+\sqrt{3}}{2}$ †) | $\frac{1+\sqrt{3}}{2}$ †) | | | |
| | | $[2\sqrt{3}-2,\ \frac{3}{2}]$ | $\frac{2}{R}$ †) | $\frac{2}{R}$ †) | | | |
| | | $[1,\ \infty)$ | $1+\frac{1}{\lceil 2R \rceil+1}$ (Thm. 17) | $1+O(\frac{\log R}{R})$ (Thm. 9) | $[\frac{3}{2},\ \infty)$ | 1 | 1 [10] |
| | gen. | $1$ | $\infty$ [19] | – | $1$ | $\infty$ [19] | – |
| | | $(1,\ \frac{3}{2}]$ | $\frac{1}{R-1}$ (Thm. 7) | $\frac{1}{R-1}$ (Thm. 12) | $(1,\ 2]$ | $\frac{1}{R-1}$ [10] | $\frac{1}{R-1}$ [10] |
| | | $[\frac{3}{2},\ 2)$ | 2 (Thm. 8) | 2 (Thm. 12) | | | |
| | | $[1,\ \infty)$ | $1+\frac{1}{R+1}$ (Thm. 6) | $1+O(\frac{\log R}{R})$ (Thm. 9) | $[2,\ \infty)$ | 1 | 1 [10] |

†) The corresponding theorems can be found in the full version [8].

## 2 Preliminaries

We denote the size and the value of an item $e$ as $s(e)$ and $v(e)$, respectively. We assume that $1 \geq s(e) > 0$ and $v(e) \geq 0$ for any $e$. For a set of items $B$, we abuse notation, and let $s(B) = \sum_{e \in B} s(e)$ and $v(B) = \sum_{e \in B} v(e)$.

For an item $e$, the ratio $v(e)/s(e)$ is called the *density* of $e$. If all the given items have the same density, we call the problem *proportional*. Without loss of generality, we assume that $v(e) = s(e)$ for the proportional case. We sometimes represent an item $e$ as the pair of its size and value $(s(e), v(e))$. Also, for the proportional case, we sometimes represent an item $e$ as its size $s(e)$.

Let $I = (e_1, \dots, e_n)$ be the input sequence of the online knapsack problem with a resource buffer. For a deterministic online algorithm ALG, let $B_i$ be the set of items in the buffer at the end of the round $i$. Note that $B_0 = \emptyset$. In the removable setting, they must satisfy $B_i \subseteq B_{i-1} \cup \{e_i\}$ and $s(B_i) \leq R$ $(i = 1, \dots, n)$. In the non-removable setting, they additionally satisfy $B_{i-1} \subseteq B_i$ $(i = 1, \dots, n)$. Without loss of generality, we assume that the algorithm transfers the optimal subset of items from the buffer into the knapsack since we do not require the online algorithm to run in polynomial time. We denote the outcome value of ALG by $\mathrm{ALG}(I)$ $(:= \max\{v(B) \mid B \subseteq B_n,\ s(B) \leq 1\})$ and the offline optimal value $\mathrm{OPT}(I)$ $(:= \max\{v(B) \mid B \subseteq \{e_1, \dots, e_n\},\ s(B) \leq 1\})$. Then, the *competitive ratio* of ALG for $I$ is defined as $\mathrm{OPT}(I)/\mathrm{ALG}(I)$ $(\geq 1)$. In addition, the competitive ratio of a problem is defined as $\inf_{\mathrm{ALG}} \sup_I \mathrm{OPT}(I)/\mathrm{ALG}(I)$, where the infimum is taken over all (deterministic) online algorithms and the supremum is taken over all input sequences.

## 3 General&Non-removable Case

To make the paper self-contained, we show that the general&non-removable case admits no constant competitive algorithm. To see this, we observe an input sequence given by Iwama and Zhang [10], which was used to prove the corresponding result for the resource augmentation setting.

▶ **Theorem 1.** *For any $R \geq 1$, there exists no constant competitive algorithm for the general&non-removable online knapsack problem with a buffer.*

**Proof.** Let $ALG$ be an online algorithm and let $R \geq 1$ and $c$ be positive reals. Consider the input sequence $I := ((1, c^1), (1, c^2), \dots, (1, c^k))$, where $(1, c^k)$ is the first item so that ALG does not take into the buffer. Note that $k \leq \lfloor R \rfloor + 1$ since the buffer size is $R$. If $k = 1$, ALG is not competitive, since $\mathrm{ALG}(I) = 0$ and $\mathrm{OPT}(I) = c$. If $k > 1$, since $\mathrm{ALG}(I) = c^{k-1}$ and $\mathrm{OPT}(I) = c^k$, the competitive ratio is $c$, which is unbounded as $c$ goes to infinity. ◀

## 4 Proportional&Non-removable Case

In this section, we consider the proportional&non-removable case. We show that the competitive ratio is $\max\{\frac{1}{R-1}, 2\}$ for the case.

### 4.1 Lower bounds

For lower bounds, we consider two cases separately: $1 < R \leq 3/2$ and $R > 3/2$.

▶ **Theorem 2.** *For all $R$ with $1 < R \leq 3/2$ and all $\epsilon > 0$, the competitive ratio of the proportional&non-removable online knapsack problem with a buffer is at least $1/(R-1) - \epsilon$.*

**Proof.** Let $\epsilon'$ be a positive real such that $\frac{1}{R-1+\epsilon'} \geq \frac{1}{R-1} - \epsilon$ and let ALG be an online algorithm. Consider the following input sequence $I$:

$$R - 1 + \epsilon', \ 1.$$

Then, ALG must pick the first item, otherwise ALG is not competitive, since $\mathrm{ALG}(I) = 0$ and $\mathrm{OPT}(I) = R - 1 + \epsilon'$. Recall that ALG cannot discard the item since we consider the non-removable setting. Also, ALG cannot take the second item since the buffer size is strictly smaller than the total size of the first and the second items. Thus, $\mathrm{ALG}(I) = R - 1 + \epsilon'$ and $\mathrm{OPT}(I) = 1$, and hence the competitive ratio is at least $\frac{1}{R-1+\epsilon'} \geq \frac{1}{R-1} - \epsilon$. ◀

It should be noted that the input sequence in the proof of Theorem 2 is the same as the one in [10], which is used to show a lower bound for the resource augmentation model.

▶ **Theorem 3.** *For all $R > 3/2$ and all $\epsilon > 0$, the competitive ratio of the proportional&non-removable online knapsack problem with a buffer is at least $2 - \epsilon$.*

**Proof.** Let $\epsilon'$ be a positive real such that $\frac{2}{1+2\epsilon'} \geq 2 - \epsilon$ and let ALG be an online algorithm. Consider the following input sequence $I$:

$$\frac{1}{2} + \epsilon', \ \frac{1}{2} + \frac{\epsilon'}{2}, \ \ldots, \ \frac{1}{2} + \frac{\epsilon'}{k}, \ \frac{1}{2} - \frac{\epsilon'}{k},$$

where the $k$th item $(1/2 + \epsilon'/k)$ is the first item that ALG does not take it into the buffer. Note that $I$ is uniquely determined by ALG and $k \leq 2R$. Since $\mathrm{ALG}(I) = 1/2 + \epsilon'$ and $\mathrm{OPT}(I) = 1/2 + \epsilon'/k + 1/2 - \epsilon'/k = 1$, the competitive ratio is at least $\frac{1}{1/2+\epsilon'} \geq 2 - \epsilon$. ◀

## 4.2 Upper bounds

For upper bounds, we consider an algorithm that greedily picks a given item if it is possible. The formal description of the algorithm is given in Algorithm 1. Recall that the resulting outcome of the algorithm is $\max\{s(B) \mid B \subseteq B_n, \ s(B) \leq 1\}$, where $B_n$ is the items in the buffer at the final round $n$. We prove that the algorithm is optimal for any $R > 1$.

**Algorithm 1** $1/(R-1)$-competitive algorithm.

```
B_0 ← ∅;
for i ← 1, 2, ... do
    if s(B_{i-1} ∪ {e_i}) ≤ R then B_i ← B_{i-1} ∪ {e_i} else B_i ← B_{i-1};
```

▶ **Theorem 4.** *Algorithm 1 is $1/(R-1)$-competitive for the proportional&non-removable online knapsack problem with a buffer when $1 < R \leq 3/2$.*

**Proof.** Let ALG be an online algorithm induced by Algorithm 1 and $I$ be an input sequence. Without loss of generality, we can assume $s(I) > R$ since otherwise $\mathrm{ALG}(I) = \mathrm{OPT}(I)$.

Suppose that $I$ does not contain items with size at least $R-1$. Let $k$ be the round such that $\sum_{i=1}^{k-1} s(e_i) < R-1 \leq \sum_{i=1}^{k} s(e_i)$. Then, we have $s(B_k) = \sum_{i=1}^{k} s(e_i) = s(e_k) + \sum_{i=1}^{k-1} s(e_i) < (R-1) + (R-1) \leq 1$ by $s(e_k) < R-1$ and $R \leq 3/2$. Therefore, in this case, the competitive ratio is at most $\frac{1}{R-1}$.

Next, suppose that $I$ contains an item with size at least $R-1$. Let $e_j$ be the first item in $I$ such that $s(e_j) \geq R-1$. If $s(B_{j-1}) \geq R-1$, then the competitive ratio is at most $\frac{1}{R-1}$ by the same argument as above. Otherwise (i.e., $s(B_{j-1}) < R-1$), we have $s(B_{j-1} \cup \{e_j\}) \leq R$ and hence $e_j \in B_j \subseteq B_n$, i.e., $e_j$ is selected in $B_n$.

Thus, $\mathrm{ALG}(I) \geq s(e_j) = R-1$ and the competitive ratio is at most $\frac{1}{R-1}$. ◀

Since $1/(R-1) = 2$ when $R = 3/2$, we obtain the following corollary from Theorem 4.

▶ **Corollary 5.** *Algorithm 1 is 2-competitive for the proportional&non-removable online knapsack problem with a buffer when $R \geq 3/2$.*

## 5 General&Removable Case

In this section, we consider the general&removable case. We show that the competitive ratio is $\max\{\frac{1}{R-1}, 2\}$ for $R \leq 2$. In addition, for general $R$, we prove that the competitive ratio is at most $1 + O(\log R/R)$ and at least $1 + \frac{1}{R+1}$.

### 5.1 Lower bounds

Here, we give lower bounds of the competitive ratio in this case. We first present a general lower bound $1 + 1/(R+1)$. The proof can be found in the full version [8].

▶ **Theorem 6.** *For $R \geq 1$, the competitive ratio of the general&removable online knapsack problem with a buffer is at least $1 + \frac{1}{R+1}$.*

Next, we provide the tight lower bound for $R \leq 2$. We separately consider the following two cases: $1 < R \leq 3/2$ and $3/2 \leq R < 2$.

▶ **Theorem 7.** *For all $R$ with $1 < R \leq 3/2$ and all $\epsilon > 0$, the competitive ratio of the general&removable online knapsack problem with a buffer is at least $1/(R-1) - \epsilon$.*

**Proof.** Let ALG be an online algorithm. Let $\hat{\epsilon}$ be a positive real such that $1/\hat{\epsilon}$ is an integer and $\min\left\{\frac{1}{(R-1+\hat{\epsilon})(1+\hat{\epsilon})}, \frac{1-\hat{\epsilon}^2}{R-1}\right\} \geq \frac{1}{R-1} - \epsilon$. In addition, let $m := 1/\hat{\epsilon}$ and $n := 1/\hat{\epsilon}^3$.

Suppose that ALG is requested the following sequence of items:

$$(1, 1),\ (\hat{\epsilon}, \hat{\epsilon}^3),\ (\hat{\epsilon}, 2\hat{\epsilon}^3),\ \ldots,\ (\hat{\epsilon}, n\hat{\epsilon}^3),$$

until ALG discards the first item $(1, 1)$. Note that the first item has a large size and a medium density, and the following items have the same small sizes but different densities that slowly increase from small to large. In addition, ALG must take the first item at the beginning (otherwise the competitive ratio becomes infinite). Thus, ALG would keep the first item and the last $\lfloor \frac{R-1}{\hat{\epsilon}} \rfloor$ items in each round.

We have two cases to consider: ALG discards the first item $(1, 1)$ or not.

**Case 1:** Suppose that ALG discards the first item $(1, 1)$ when the item $(\hat{\epsilon}, i\hat{\epsilon}^3)$ comes. Note that the requested sequence is $I := \big((1, 1), (\hat{\epsilon}, \hat{\epsilon}^3), (\hat{\epsilon}, 2\hat{\epsilon}^3), \ldots, (\hat{\epsilon}, i\hat{\epsilon}^3)\big)$. Then, we have $\mathrm{ALG}(I) \leq (\lfloor \frac{R-1}{\hat{\epsilon}} \rfloor + 1)i\hat{\epsilon}^3$ (since ALG keeps at most $\lfloor \frac{R-1}{\hat{\epsilon}} \rfloor + 1$ small items at the end) and $\mathrm{OPT}(I) \geq \max\{1,\, m \cdot (i-m)\hat{\epsilon}^3\}$ (the left term 1 comes from the first item and the right term $m \cdot (i-m)\hat{\epsilon}^3$ comes from the last $m$ $(= 1/\hat{\epsilon})$ items). Hence, the competitive ratio is at least $\frac{\max\{1,\, m\cdot(i-m)\hat{\epsilon}^3\}}{(\lfloor \frac{R-1}{\hat{\epsilon}} \rfloor+1)i\hat{\epsilon}^3} \geq \frac{1}{(R-1+\hat{\epsilon})(1+\hat{\epsilon})} \geq \frac{1}{R-1} - \epsilon$.

**Case 2:** Suppose that ALG does not reject the first item until the end. Then, the competitive ratio is at least $\frac{m\cdot(n-m)\hat{\epsilon}^3}{\lfloor \frac{R-1}{\hat{\epsilon}} \rfloor n\hat{\epsilon}^3} \geq \frac{1}{R-1} \cdot \frac{m(n-m)\hat{\epsilon}^3}{n\hat{\epsilon}^2} = \frac{1-\hat{\epsilon}^2}{R-1} \geq \frac{1}{R-1} - \epsilon$. ◀

▶ **Theorem 8.** *For all $R$ with $3/2 \leq R < 2$ and all $\epsilon > 0$, the competitive ratio of the general&removable online knapsack problem with a buffer is at least $2 - \epsilon$.*

**Proof.** Let $k$ be an integer such that $k > \max\{\frac{1}{2-R}, \frac{1}{\epsilon}\}$. Let ALG be an online algorithm.

Consider the item sequence $I \coloneqq (e_1, \ldots, e_k)$ where $(s(e_i), v(e_i)) = (1 - \frac{i}{2k^2}, 1 - \frac{i}{2k})$ for $i = 1, \ldots, k$. Then, at the end of the sequence, ALG must keep exactly one item because it must select at least one item (otherwise the competitive ratio is unbounded) and every pair of items exceeds the capacity of the buffer (i.e., $s(e_i) + s(e_j) \geq 2(1 - \frac{k}{2k^2}) = 2 - \frac{1}{k} > R$ for any $i, j \in \{1, \ldots, k\}$).

Suppose that $\{e_i\}$ is selected in the buffer at the end of the sequence $I$. If $i = k$, then the competitive ratio for $I$ is $\frac{\mathrm{OPT}(I)}{\mathrm{ALG}(I)} = \frac{v(e_1)}{v(e_k)} = \frac{1-\frac{1}{2k}}{1-\frac{1}{2}} = 2 - \frac{1}{k} > 2 - \epsilon$. Otherwise (i.e., $i < k$), let us consider a sequence $I' \coloneqq (e_1, \ldots, e_k, e_{k+1})$ with $(s(e_{k+1}), v(e_{k+1})) = (\frac{i+1}{2k^2}, 1 - \frac{i}{2k})$. Then, the competitive ratio for $I'$ is at least $\frac{\mathrm{OPT}(I')}{\mathrm{ALG}(I')} = \frac{v(e_{i+1})+v(e_{k+1})}{v(e_i)} = \frac{(1-\frac{i+1}{2k})+(1-\frac{i}{2k})}{1-\frac{i}{2k}} = 2 - \frac{1}{2k-i} \geq 2 - \frac{1}{k} > 2 - \epsilon$. ◀

## 5.2 Upper bounds

Here, we provide an asymptotically nearly optimal algorithm for large $R$ and an optimal algorithm for small $R$ ($< 2$).

First, we provide a $(1 + O(\log R/R))$-competitive algorithm for the asymptotic case. Suppose that $R$ is sufficiently large. Let $m \coloneqq \lfloor (R-3)/2 \rfloor$ and let $\epsilon$ ($\leq 1$) be a positive real such that $\log_{1+\epsilon}(1/\epsilon) = m$. Note that we have $m = \Theta(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ and $\epsilon = O(\log R/R)$ (see Lemma 18 in Appendix A).

We partition all the items as follows. Let $S$ be the set of items with size at most $\epsilon$. Let $M$ be the set of items not in $S$ and let $M^j$ ($j \in \mathbb{Z}$) be the set of items $e \in M$ with $(1+\epsilon)^j \leq v(e) < (1+\epsilon)^{j+1}$ (note that $j$ is not restricted to be positive). Let us consider Algorithm 2 for the problem. Intuitively, the algorithm selects items in greedy ways for $S$ and each $M^j$ with $\nu_i \leq j \leq \mu_i$. Note that for any $i \geq 1$, we have $\mu_i - \nu_i = 2m$. For each $i \geq 1$, since $s(B_i \cap S) \leq 2 + \epsilon$ and $s(B_i \cap M^j) \leq 1$ for any $\nu_i \leq j \leq \mu_i$, we have $s(B_i) \leq 2m + 2 + \epsilon \leq R$. Thus, the algorithm is applicable.

**Algorithm 2** $(1 + O(\log R/R))$-competitive algorithm.

$B_0 \leftarrow \emptyset$;
**for** $i \leftarrow 1, 2, \ldots$ **do**
  $B_i \leftarrow \emptyset$ and $B'_i \leftarrow (B_{i-1} \cup \{e_i\})$;
  **foreach** $e \in B'_i \cap S$ *in the non-increasing order of the density* **do**
    $B_i \leftarrow B_i \cup \{e\}$;
    **if** $s(B_i) > 2$ **then break**;
  Let $e_i^* \in \arg\max\{v(e) \mid e \in B'_i\}$;
  Let $\mu_i \leftarrow \lfloor \log_{1+\epsilon} v(e_i^*) \rfloor$ and $\nu_i \leftarrow \lfloor \log_{1+\epsilon} \epsilon^2 v(e_i^*) \rfloor$; // $e_i^* \in M^{\mu_i}$
  **for** $j \leftarrow \nu_i, \ldots, \mu_i$ **do**
    **foreach** $e \in B'_i \cap M^j$ *in the non-decreasing order of the size* **do**
      **if** $s(B_i \cap M^j) + s(e) \leq 1$ **then** $B_i \leftarrow B_i \cup \{e\}$;

▶ **Theorem 9.** *Algorithm 2 is $(1+O(\log R/R))$-competitive for the general&removable online knapsack problem with a buffer when $R$ is a sufficiently large real.*

Let $I := (e_1, \dots, e_n)$ be an input sequence, $B_{\mathrm{OPT}} \in \arg\max\{v(X) \mid s(X) \leq 1,\ X \subseteq \{e_1, \dots, e_n\}\}$ be the offline optimal solution, and $B_{\mathrm{ALG}} \in \arg\max\{v(X) \mid s(X) \leq 1,\ X \subseteq B_n\}$ be the outcome solution of ALG. We construct another feasible solution $B^*$ from $B_n$ by Algorithm 3. Note that $v(B_{\mathrm{ALG}}) \geq v(B^*)$.

**Algorithm 3** Construct a feasible solution.

$B^* \leftarrow B_n \cap B_{\mathrm{OPT}}$;
**for** $k \leftarrow \nu_n, \dots, \mu_n$ **do**
  Let $r_k \leftarrow |(B_{\mathrm{OPT}} \setminus B^*) \cap M^k|$;
  **for** $j \leftarrow 1, \dots, r_k$ **do**
    Let $a \leftarrow \arg\min\{s(e) \mid e \in (B_n \setminus B^*) \cap M^k\}$ and $B^* \leftarrow B^* \cup \{a\}$;
**while** $(B_n \setminus B^*) \cap S \neq \emptyset$ **do**
  Let $a \in \arg\max\{v(e)/s(e) \mid e \in (B_n \setminus B^*) \cap S\}$;
  **if** $s(B^*) + s(a) \leq 1$ **then** $B^* \leftarrow B^* \cup \{a\}$;
  **else break**;
**return** $B^*$;

To prove the theorem, we show the following two claims.

▷ Claim 10. $v(B_{\mathrm{OPT}} \cap M) \leq (1+\epsilon)v(B^* \cap M) + \epsilon v(B_{\mathrm{OPT}})$ and $s(B_{\mathrm{OPT}} \cap M) \geq s(B^* \cap M)$.

▷ Claim 11. $v(B_{\mathrm{OPT}} \cap S) \leq v(B^* \cap S) + \epsilon(1+2\epsilon)v(B_{\mathrm{OPT}})$.

With these claims, $B^*$ is feasible, and we have $v(B_{\mathrm{OPT}}) = v(B_{\mathrm{OPT}} \cap M) + v(B_{\mathrm{OPT}} \cap S) \leq (1+\epsilon)v(B^*) + (2\epsilon + 2\epsilon^2)v(B_{\mathrm{OPT}})$. This implies $(1 - 2\epsilon - 2\epsilon^2)v(B_{\mathrm{OPT}}) \leq (1+\epsilon)v(B^*)$. Since $v(B^*) \leq v(B_{\mathrm{ALG}})$, the competitive ratio of Algorithm 2 is at most $\frac{1+\epsilon}{1-2\epsilon-2\epsilon^2} \leq \frac{1+\epsilon}{1-3\epsilon} \leq 1 + 6\epsilon = 1 + O(\log R/R)$, when $\epsilon < 1/12$ (this inequality follows from the assumption that $R$ is sufficiently large).

The proof is completed by proving Claims 10 and 11.

Proof of Claim 10. Note that $v(B_{\mathrm{OPT}} \cap M) = \sum_{k<\nu_n} v(B_{\mathrm{OPT}} \cap M^k) + \sum_{k\geq\nu_n} v(B_{\mathrm{OPT}} \cap M^k)$. For $e \in M^k$ with $k < \nu_n$, we have $s(e) > \epsilon$ and $v(e) < (1+\epsilon)^{\nu_n} \leq \epsilon^2 v(e_n^*)$, and hence $v(e)/s(e) \leq \epsilon^2 v(e_n^*)/\epsilon \leq \epsilon v(B_{\mathrm{OPT}})$. Thus, we have $\sum_{k<\nu_n} v(B_{\mathrm{OPT}} \cap M^k) \leq \epsilon v(B_{\mathrm{OPT}})$. For $k$ with $\mu_n \leq k \leq \nu_n$, the set $B_n \cap M^k$ is the greedy solution for $M^k$ according to the non-decreasing order of their size. Hence, by the construction of $B^*$, the number of items in $B_{\mathrm{OPT}} \cap M^k$ equals to the number of items in $B^* \cap M^k$, and we have $s(B_{\mathrm{OPT}} \cap M^k) \geq s(B^* \cap M^k)$. Also, for each $e \in B_{\mathrm{OPT}} \cap M^k$ and $f \in B^* \cap M^k$, $v(e)/v(f) < (1+\epsilon)^{k+1}/(1+\epsilon)^k = (1+\epsilon)$. Hence, $\sum_{k\geq\nu_n} v(B_{\mathrm{OPT}} \cap M^k) \leq (1+\epsilon)\sum_{k\geq\nu_n} v(B^* \cap M^k)$. ◁

Proof of Claim 11. It is sufficient to consider the case $B_{\mathrm{OPT}} \cap S \not\subseteq B_n$, since otherwise $B_{\mathrm{OPT}} \cap S \subseteq B^* \cap S$ and the claim clearly holds. Hence, we have $s(B_n \cap S) > 2$. Let $B_n \cap S = \{f_1, f_2, \dots, f_{|B_n \cap S|}\}$ be sorted in non-increasing order of their density. Let $f_j$ be the item with the largest index in $(B_n \cap S) \setminus B_{\mathrm{OPT}}$. Also let $\ell \geq 1$ be the index such that $\sum_{i=1}^{\ell} s(f_i) \leq 1 < \sum_{i=1}^{\ell+1} s(f_i)$. There are two cases to consider: $j \leq \ell$ and $j > \ell$.

**Case 1:** Suppose that $j \leq \ell$. Then, by the definition of $f_j$, we have $\{f_{\ell+1}, \dots f_{|B_n \cap S|}\} \subseteq B_{\mathrm{OPT}}$. Since $s(B_n \cap S) > 2$, we have $s(B_{\mathrm{OPT}}) \geq s(B_{\mathrm{OPT}} \cap S) \geq s(B_n \cap S) - \sum_{i=1}^{\ell} s(f_i) > 1$, which contradicts with $s(B_{\mathrm{OPT}}) \leq 1$.

**Case 2:** Suppose that $j > \ell$. In this case, we prove that $v(f_j) \le \epsilon(1+2\epsilon)v(B_{\mathrm{OPT}})$. Since $\sum_{i=1}^{\ell} s(f_i) \ge 1-\epsilon$, we have $(1-\epsilon)\cdot\frac{v(f_j)}{s(f_j)} \le \sum_{i=1}^{\ell} s(f_i)\cdot\frac{v(f_j)}{s(f_j)} \le \sum_{i=1}^{\ell} s(f_i)\cdot\frac{v(f_i)}{s(f_i)} = \sum_{i=1}^{\ell} v(f_i) \le v(B_{\mathrm{OPT}})$. Therefore, $v(f_j) \le \frac{s(f_j)}{1-\epsilon}v(B_{\mathrm{OPT}}) \le \frac{\epsilon}{1-\epsilon}v(B_{\mathrm{OPT}}) \le \epsilon(1+2\epsilon)v(B_{\mathrm{OPT}})$ when $\epsilon \le 1/2$.
Since $s(B^* \cap M) \le s(B_{\mathrm{OPT}} \cap M)$ by construction of $B^*$, we have $s(B^* \cap S) + s(f_j) \ge s(B_{\mathrm{OPT}} \cap S)$. By construction of $B_n \cap S$, we have $\min\{v(f)/s(f) \mid f \in (B_n \cap S)\setminus B_{\mathrm{OPT}}\} \ge \max\{v(f)/s(f) \mid f \in (B_{\mathrm{OPT}} \cap S)\setminus B_n\}$. Therefore, $v(B^* \cap S) + v(f_j) \ge v(B_{\mathrm{OPT}} \cap S)$. Moreover we have $v(f_j) \le \epsilon(1+2\epsilon)v(B_{\mathrm{OPT}})$, and the claim follows. ◁

The proof of Theorem 9 is completed.

Next, let us consider an algorithm that selects items according to the non-increasing order of the density. The algorithm is formally described in Algorithm 4. We prove that it is optimal when $1 < R < 2$.

**Algorithm 4** $\max\{1/(R-1), 2\}$-competitive algorithm for $1 < R < 2$.

$B_0 \leftarrow \emptyset$;
**for** $i \leftarrow 1, 2, \ldots$ **do**
  $B_i \leftarrow \emptyset$;
  **foreach** $e \in B_{i-1} \cup \{e_i\}$ *in the non-increasing order of the density* **do**
    **if** $s(B_i) + s(e) \le R$ **then** $B_i \leftarrow B_i \cup \{e\}$ ;

▶ **Theorem 12.** *Algorithm 4 is* $\max\{1/(R-1), 2\}$*-competitive for the general&removable online knapsack problem with a buffer when* $1 < R < 2$.

**Proof.** Let $I := (e_1, \ldots, e_n)$ be an input sequence. Without loss of generality, we can assume that $\sum_{i=1}^{n} s(e_i) > R$ since otherwise $\mathrm{ALG}(I) = \mathrm{OPT}(I)$. Let $f_1, \ldots, f_n$ be the rearrangement of $I$ according to the non-increasing order of the density, i.e., $\{f_1, \ldots, f_n\} = \{e_1, \ldots, e_n\}$ and $v(f_1)/s(f_1) \ge \cdots \ge v(f_n)/s(f_n)$. Let $k$ $(\le n-1)$ be the index such that $\sum_{i=1}^{k} s(f_i) \le 1 < \sum_{i=1}^{k+1} s(f_i)$. Then, by the definition of the algorithm, we have $\{f_1, \ldots, f_k\} \subseteq B_n$. There are two cases to consider: $f_{k+1} \notin B_n$ and $f_{k+1} \in B_n$.

**Case 1:** Suppose that $f_{k+1} \notin B_n$. Then, we have $\sum_{i=1}^{k+1} s(f_i) > R$, and hence $\sum_{i=1}^{k} s(f_i) > R - s(f_{k+1}) \ge R - 1$ by $s(f_{k+1}) \le 1$. Thus, $\mathrm{OPT}(I)$ is at most $\mathrm{ALG}(I)/(R-1)$ and the competitive ratio is at most $1/(R-1)$.

**Case 2:** Suppose that $f_{k+1} \in B_n$. By a similar analysis of the famous 2-approximation algorithm for the offline knapsack problem, we have $\mathrm{OPT}(I) \le \sum_{i=1}^{k} v(f_i) + v(f_{k+1}) \le 2\cdot\max\{\sum_{i=1}^{k} v(f_i), v(f_{k+1})\} \le 2\cdot\mathrm{ALG}(I)$. Thus, the competitive ratio is at most 2. ◀

## 6 Proportional&Removable Case

In this section, we consider the proportional&removable case. We consider the following four cases separately: (i) $1 \le R \le \frac{1+\sqrt{2}}{2}$, (ii) $2-\frac{\sqrt{2}}{2} \le R \le 17-9\sqrt{3}$, (iii) $2\sqrt{3}-2 \le R \le 3/2$, and (iv) general $R$ (see Figure 1). We remark that the competitive ratios for $\frac{1+\sqrt{2}}{2} \le R \le 2-\frac{\sqrt{2}}{2}$ (and $17-9\sqrt{3} \le R \le 2\sqrt{3}-2$) can be obtained by considering the upper bound for $R = \frac{1+\sqrt{2}}{2}$ in case (i) ($R = 17-9\sqrt{3}$ in case (ii)) and the lower bound for $R = 2-\frac{\sqrt{2}}{2}$ in case (ii) ($R = 2\sqrt{3}-2$ in case (iii)). Due to space limitation, we only analyze cases (i) and (iv). The analysis for (ii) and (iii) can be found in the full version [8].

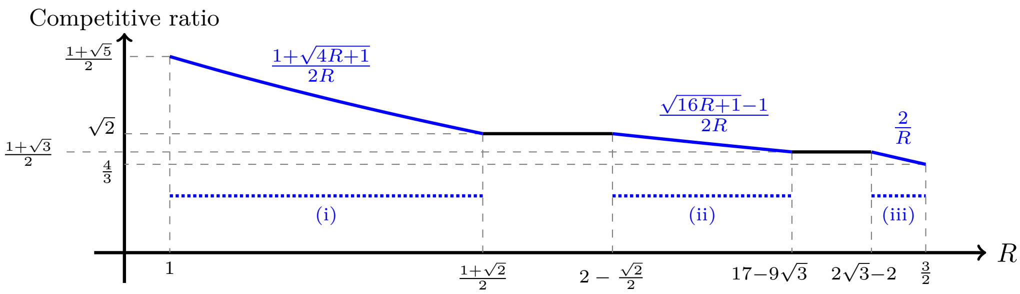


**Figure 1** The competitive ratios for the proportional&removable case with $1 \leq R \leq \frac{3}{2}$.

## 6.1 $1 \leq R \leq \frac{1+\sqrt{2}}{2}$

We prove that the competitive ratio is $\frac{1+\sqrt{4R+1}}{2R}$ when $1 \leq R \leq \frac{1+\sqrt{2}}{2}$. Let $r > 0$ be a real such that $r + r^2 = R$, i.e., $r = \frac{\sqrt{1+4R}-1}{2}$.

### 6.1.1 Lower bound

We first prove the lower bound.

▶ **Theorem 13.** *For any $\epsilon > 0$, the competitive ratio of the proportional&removable online knapsack problem with a buffer is at least $\frac{1+\sqrt{4R+1}}{2R} - \epsilon$ when $1 \leq R < 2$.*

**Proof.** Let ALG be an online algorithm and let $\epsilon'$ be a positive real such that $\frac{r}{r^2+\epsilon'} \geq \frac{1}{r} - \epsilon$ and $\epsilon' < r - r^2$. Note that $r = \frac{\sqrt{1+4R}-1}{2} < 1$ and $\frac{1}{r} = \frac{1+\sqrt{4R+1}}{2R}$. Consider the input sequence $I \coloneqq (e_1, e_2)$ where $s(e_1) = r$ and $s(e_2) = r^2 + \epsilon'$. Since $r + r^2 = R$, ALG must discard at least one of them. If ALG discards the item with size $r$, then the competitive ratio for the sequence is $\frac{r}{r^2+\epsilon'} \geq \frac{1}{r} - \epsilon = \frac{1+\sqrt{4R+1}}{2R} - \epsilon$. If ALG discards the item with size $r^2 + \epsilon'$, let $I' \coloneqq (e_1, e_2, e_3)$ where $s(e_3) = 1 - r^2 - \epsilon'$. As $r \geq 1 - r^2$ and $r + (1 - r^2 - \epsilon') > 1$, we have $\mathrm{OPT}(I') = 1$ and $\mathrm{ALG}(I') \leq r$. Hence the competitive ratio is at least $\frac{1}{r} = \frac{1+\sqrt{4R+1}}{2R}$. ◀

### 6.1.2 Upper bound for $1 \leq R \leq 10/9$

Next, we give an optimal algorithm for $1 \leq R \leq 10/9$. In this subsubsection, an item $e$ is called *small*, *medium*, and *large* if $s(e) \leq r^2$, $r^2 < s(e) < r$, and $r \leq s(e)$, respectively. Let $S$, $M$, and $L$ respectively denote the sets of small, medium, and large items.

We consider Algorithm 5, which is a generalization of the $\frac{1+\sqrt{5}}{2}$-competitive algorithm for $R = 1$ given by Iwama and Taketomi [9]. If the algorithm can select a set of items $B'$ such that $r \leq s(B') \leq 1$, it keeps the set $B'$ until the end since it is sufficient to achieve $1/r$-competitive. Otherwise, it picks the smallest medium item (if exists) and greedily selects small items according to the non-increasing order of the sizes. We show that it is optimal when $1 \leq R \leq 10/9$.

▶ **Theorem 14.** *Algorithm 5 is $\frac{1+\sqrt{1+4R}}{2R}$-competitive for the proportional&removable online knapsack problem with a buffer when $1 \leq R \leq 10/9$.*

**Proof.** Let $I \coloneqq (e_1, \ldots, e_n)$ be the input sequence. If there exists a large item $e_i$, the competitive ratio is at most $1/r = \frac{1+\sqrt{1+4R}}{2R}$ by $r \leq s(e_i) \leq 1$. If there exist two medium items $e_i, e_j$ such that $s(e_i) + s(e_j) \leq 1$, the competitive ratio is at most $1/r = \frac{1+\sqrt{1+4R}}{2R}$ by $r < 2r^2 < s(e_i) + s(e_j) \leq 1$. In what follows, we assume that all the input items are

**Algorithm 5** $\frac{1+\sqrt{1+4R}}{2R}$-competitive algorithm for $1 \le R \le \frac{10}{9}$.

$B_0 \leftarrow \emptyset$;
**for** $i \leftarrow 1, 2, \ldots$ **do**
  **if** $\exists B' \subseteq B_{i-1} \cup \{e_i\}$ *such that* $r \le s(B') \le 1$ **then** $B_i \leftarrow B'$;
  **else if** $e_i \in M$ *and* $|B_{i-1} \cap M| = 1$ **then**
    let $\{e'_i\} = B_{i-1} \cap M$;
    **if** $s(e_i) < s(e'_i)$ **then** $B_i \leftarrow B_{i-1} \cup \{e_i\} \setminus \{e'_i\}$;
    **else** $B_i \leftarrow B_{i-1}$;
  **else**
    $B_i \leftarrow \emptyset$;
    **foreach** $e \in B_{i-1} \cup \{e_i\}$ *in non-increasing order of size* **do**
      **if** $s(B_i) + s(e) \le R$ **then** $B_i \leftarrow B_i \cup \{e\}$;

not large and every pair of medium items cannot be packed into the knapsack together. In addition, suppose that $s(B_n) \notin [r, 1] > 1$ since otherwise the competitive ratio is at most $1/r = \frac{1+\sqrt{1+4R}}{2R}$. By the algorithm, this additional assumption means $s(B') \notin [1, r]$ for any $B' \subseteq B_{i-1} \cup \{e_i\}$ with $i \in \{1, \ldots, n\}$.

If $\{e_1, \ldots, e_n\} \cap S \subseteq B_n$, the competitive ratio is at most

$$\frac{r + s(\{e_1, \ldots, e_n\} \cap S)}{r^2 + s(\{e_1, \ldots, e_n\} \cap S)} \le \frac{1}{r} = \frac{1+\sqrt{1+4R}}{2R}.$$

Otherwise, i.e., $\{e_1, \ldots, e_n\} \cap S \not\subseteq B_n$, let $e_i$ be a small item that is not in $B_n$, and $j$ be the smallest index such that $j \ge i$ and $e_i \notin B_j$. Note that $e_i \in B_{j-1} \cup \{e_j\}$. We have four cases to consider.

**Case 1:** Suppose that $s(e_i) \ge r/2$. In this case, there exists $e' \in B_j$ such that $r^2 \ge s(e') \ge s(e_i)$. Thus, we have $r \le s(e_i) + s(e') \le 1$, a contradiction.

**Case 2:** Suppose that there exists no medium item in $B_j$. Then, there exists $B' \subseteq B_{j-1} \cup \{e_j\}$ such that $r \le s(B_j) \le 1$, because $s(B_{j-1} \cup \{e_j\}) > R$ and all the items in $B_{j-1} \cup \{e_j\}$ are small. This is a contradiction.

**Case 3:** Suppose that $s(e) < r/2$ for any $e \in B_j \cap S$. Then, we have $r \le s(B_j) \le 1$, a contradiction.

**Case 4:** Let us consider the other case, i.e., $s(e_i) < r/2$, $\exists e \in B_j \cap M$, and $\exists e' \in B_j \cap S$ such that $s(e') \ge r/2$. Then, $s(B_j) - s(e) + s(e_i) \ge R - s(e) \ge R - r^2 = r$. Also, $s(B_j) - s(e') \le R - s(e') \le R - r/2 = r^2 + r/2 \le 1$. By the additional assumption, we have $s(B_j) - s(e) + s(e_i) > 1$ and $s(B_j) - s(e') < r$. Thus, we have $s(B_j) > 1 + s(e) - s(e_i) > 1 + r^2 - r/2 = (1-r)^2 + 3r/2 \ge r + r/2 \ge r + s(e') > s(B_j)$, which is a contradiction. ◀

### 6.1.3 Upper bound for $\frac{10}{9} \le R \le \frac{1+\sqrt{2}}{2}$

Recall that $r > 0$ is a real such that $r + r^2 = R$, i.e., $r = \frac{\sqrt{1+4R}-1}{2}$. For $\frac{10}{9} \le R \le \frac{1+\sqrt{2}}{2}$, we have $2/3 \le r \le 1/\sqrt{2}$ and $1 - r \le r/2 \le r^2 \le 1/2 < r < 1$. In this subsubsection, an item $e$ is called *small*, *medium*, and *large* if $s(e) \le 1 - r$, $1 - r < s(e) < r$, and $r \le s(e)$, respectively. Let $S$, $M$, and $L$ respectively denote the sets of small, medium, and large items. In addition, $M$ is further partitioned into three subsets $M_i$ $(i = 1, 2, 3)$, where $M_1$, $M_2$, $M_3$ respectively denote the set of the items $e$ with size $1 - r < s(e) \le r/2$, $r/2 < s(e) < r^2$, and $r^2 \le s(e) < r$.

We consider Algorithm 6 for the problem. If the algorithm can select a set of items $B'$ such that $r \le s(B') \le 1$, it keeps the set $B'$ until the end. Otherwise, it partitions the buffer into two spaces with size $r$ and $r^2$. All the small items are taken into the first space. If the set of medium items is of size at least $r^2$, then the smallest its subset $B'$ with size at least $r^2$ is selected into the first space. If the set of medium items is of size at most $r^2$, then all of them are selected into the first space. If there are remaining medium items, the smallest one is kept in the second space if its size is smaller than $r^2$. We show that the algorithm is optimal when $\frac{10}{9} \le R \le \frac{1+\sqrt{2}}{2}$.

**Algorithm 6** $\frac{1+\sqrt{1+4R}}{2R}$-competitive algorithm for $\frac{10}{9} \le R \le \frac{1+\sqrt{2}}{2}$.

$B_0 \leftarrow \emptyset$, $B_0^{(1)} \leftarrow \emptyset$, $B_0^{(2)} \leftarrow \emptyset$;
**for** $i \leftarrow 1, 2, \dots$ **do**
  **if** $\exists B' \subseteq B_{i-1} \cup \{e_i\}$ *such that* $r \le s(B') \le 1$ **then** $B_i^{(1)} \leftarrow B'$ and $B_i^{(2)} \leftarrow \emptyset$ ;
  **else if** $s((B_{i-1} \cup \{e_i\}) \cap M) \ge r^2$ **then**
    let $T_i \in \arg\min\{s(B') \mid B' \subseteq (B_{i-1} \cup \{e_i\}) \cap M,\ s(B') \ge r^2\}$;
    $B_i^{(1)} \leftarrow T_i \cup ((B_{i-1} \cup \{e_i\}) \cap S)$;
    **if** $B_i^{(1)} \ne B_{i-1} \cup \{e_i\}$ **then**
      let $a \in \arg\min\{s(e) \mid e \in B_{i-1} \cup \{e_i\} \setminus B_i^{(1)}\}$;
      **if** $a \in M_1 \cup M_2$ **then** $B_i^{(2)} \leftarrow \{a\}$;
  **else** $B_i^{(1)} \leftarrow B_{i-1} \cup \{e_i\}$ and $B_i^{(2)} \leftarrow \emptyset$ ;
  $B_i \leftarrow B_i^{(1)} \cup B_i^{(2)}$;

▶ **Theorem 15.** *Algorithm 6 is $\frac{1+\sqrt{1+4R}}{2R}$-competitive for the proportional&removable online knapsack problem with a buffer when $10/9 \le R \le \frac{1+\sqrt{2}}{2}$.*

Let $I := (e_1, \dots, e_n)$ be the input sequence and let $I_k := \{e_1, \dots, e_k\}$ be the first $k$ items of $I$.

▶ **Lemma 16.** *If $I_n \subseteq M$, then Algorithm 6 is $\frac{1+\sqrt{1+4R}}{2R}$ $(= 1/r)$-competitive when $10/9 \le R \le \frac{1+\sqrt{2}}{2}$.*

**Proof.** The proof can be found in the full version [8]. ◀

Now, we are ready to prove Theorem 15.

**Proof of Theorem 15.** Let $\mathrm{OPT} \in \arg\max\{s(X) \mid X \subseteq I_n,\ s(X) \le 1\}$ and $\mathrm{OPT}_M \in \arg\max\{s(X) \mid X \subseteq I_n \cap M,\ s(X) \le 1\}$. Without loss of generality, we can assume that $\sum_{i=1}^{n} s(e_i) > R$.

If $e_i \in L$ for some $i$, then $r \le s(B_n^{(1)}) \le 1$. Thus, we assume that all the items in the input sequence are not large, i.e., $I_n \cap L = \emptyset$.

Suppose that Algorithm 6 discards some small items, i.e., $I_n \cap S \ne B_n \cap S$. Let $j$ be the round such that $I_{j-1} \cap S = B_{j-1} \cap S$ and $I_j \cap S \ne B_j \cap S$. Let $T_j \in \arg\min\{s(B') \mid B' \subseteq (B_{j-1} \cup \{e_j\}) \cap M,\ s(B') > r\}$. Since $I_{j-1} \cap S = B_{j-1} \cap S$ and $I_j \cap S \ne B_j \cap S$, we have $s(T_j \cup (I_j \cap S)) > 1$. Since $s(e) < 1 - r$ $(\forall e \in S)$, there exists $S' \in I_j \cap S$ such that $r \le s(T_j \cup S') \le 1$. Therefore, if $I_n \cap S \ne B_n \cap S$, then $\mathrm{ALG}(I) \ge r$.

Consequently, we assume $I_n \cap L = \emptyset$ and $I_n \cap S \subseteq B_n$. Then, the competitive ratio is at most

$$\frac{s(\mathrm{OPT})}{s(B_n^{(1)})} \leq \frac{s(\mathrm{OPT}_M) + s(I_n \cap S)}{s(B_n^{(1)} \cap M) + s(I_n \cap S)} \leq \frac{s(\mathrm{OPT}_M)}{s(B_n^{(1)} \cap M)},$$

and hence we can assume, without loss of generality, that $I_n \subseteq M$.

Thus, by Lemma 16, the theorem is proved. ◀

## 6.2 General $R$

In this subsection, we consider proportional&removable case with general $R$. By Theorem 9, the upper bound of the competitive ratio is $1 + O(\log R/R)$. Hence, we only give a lower bound of the competitive ratio.

▶ **Theorem 17.** *For any positive real $\epsilon < 1$, the competitive ratio of the proportional&removable online knapsack problem with a buffer is at least $1 + \frac{1}{\lceil 2R \rceil + 1} - \epsilon$.*

**Proof.** Let $n := \lceil 2R \rceil + 1$ and let ALG be an online algorithm. Consider the item sequence $I := (e_1, \dots, e_{n-1}, e_n)$ where $s(e_i) = \frac{i}{n} + \frac{\epsilon}{n^2}$ for $i = 1, \dots, n-1$ (we will set $s(e_n)$ later depending on ALG). At the end of $(n-1)$st round, ALG must discard at least one item because $\sum_{i=1}^{n-1} s(e_i) > \frac{n-1}{2} = \frac{\lceil 2R \rceil}{2} \geq R$. Suppose that ALG discards $e_j$, and let $s(e_n) = 1 - s(e_j)$. Then, we have $\mathrm{OPT}(I) = s(e_j) + s(e_n) = 1$. We will prove that $\mathrm{ALG}(I)$ is at most $1 - (1-\epsilon)/n$, which implies that the competitive ratio of ALG is at least $\frac{1}{1-(1-\epsilon)/n} \geq 1 + (1-\epsilon)/n \geq 1 + \frac{1}{\lceil 2R \rceil + 1} - \epsilon$.

Let $B^*$ be the output of ALG, i.e., $s(B^*) = \mathrm{ALG}(I)$. We have two cases to consider: $e_n \notin B^*$ and $e_n \in B^*$.

**Case 1:** If $e_n \notin B^*$, then we have $s(B^*) = \frac{\sum_{e_i \in B^*} i}{n} + \frac{|B^*| \cdot \epsilon}{n^2}$. We assume $B^* \neq \emptyset$ since otherwise $s(B^*) = 0$. Since $s(B^*) \leq 1$ and $\frac{|B^*| \cdot \epsilon}{n^2} > 0$, we have $\frac{\sum\{i | e_i \in B^*\}}{n} \leq \frac{n-1}{n}$. Hence, we obtain $s(B^*) \leq \frac{n-1}{n} + \frac{n \cdot \epsilon}{n^2} = 1 - \frac{1-\epsilon}{n}$.

**Case 2:** If $e_n \in B^*$, then we have $s(B^*) = \frac{(n-j) + \sum\{i | e_i \in B^* \setminus \{e_n\}\}}{n} + \frac{(|B^*|-2) \cdot \epsilon}{n^2}$. We assume $|B^*| \geq 3$ since otherwise $s(B^*) \leq \frac{n-1}{n}$ by $e_j \notin B^*$. Since $s(B^*) \leq 1$ and $\frac{(|B^*|-2) \cdot \epsilon}{n^2} > 0$, we have $\frac{(n-j) + \sum\{i | e_i \in B^* \setminus \{e_n\}\}}{n} \leq \frac{n-1}{n}$. Hence, we obtain $s(B^*) \leq \frac{n-1}{n} + \frac{n \cdot \epsilon}{n^2} = 1 - \frac{1-\epsilon}{n}$. ◀

## References


1 Susanne Albers, Arindam Khan, and Leon Ladewig. Improved Online Algorithms for Knapsack and GAP in the Random Order Model. In *Proceedings of APPROX/RANDOM*, pages 22:1–22:23, 2019.

2 Badanidiyuru Ashwinkumar and Robert Kleinberg. Randomized Online Algorithms for the Buyback Problem. In *Internet and Network Economics*, pages 529–536, 2009.

3 Moshe Babaioff, Jason D. Hartline, and Robert D. Kleinberg. Selling Ad Campaigns: Online Algorithms with Cancellations. In *Proceedings of EC*, 2009.

4 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *Proceedings of APPROX/RANDOM*, pages 16–28. Springer, 2007.

5 Marek Cygan, Łukasz Jeż, and Jiří Sgall. Online Knapsack Revisited. *Theory of Computing Systems*, 58(1):153–190, 2016.

6 Xin Han, Yasushi Kawase, and Kazuhisa Makino. Online Unweighted Knapsack Problem with Removal Cost. *Algorithmica*, 70(1):76–91, 2014.

7 Xin Han, Yasushi Kawase, and Kazuhisa Makino. Randomized algorithms for online knapsack problems. *Theoretical Computer Science*, 562:395–405, 2015.
8 Xin Han, Yasushi Kawase, Kazuhisa Makino, and Haruki Yokomaku. Online Knapsack Problems with a Resource Buffer. *CoRR*, abs/1909.10016, 2019. arXiv:1909.10016.
9 Kazuo Iwama and Shiro Taketomi. Removable Online Knapsack Problems. In *Proceeding of ICALP*, pages 293–305, 2002.
10 Kazuo Iwama and Guochuan Zhang. Optimal Resource Augmentations for Online Knapsack. In *Proceedings of APPROX/RANDOM*, pages 180–188, 2007.
11 Yasushi Kawase, Xin Han, and Kazuhisa Makino. Proportional cost buyback problem with weight bounds. *Theoretical Computer Science*, 2016.
12 Yasushi Kawase, Xin Han, and Kazuhisa Makino. Unit Cost Buyback Problem. *Theory of Computing Systems*, 2018.
13 Yasushi Kawase and Atsushi Iwasaki. Near-Feasible Stable Matchings with Budget Constraints. In *Proceedings of IJCAI*, pages 242–248, 2017.
14 Yasushi Kawase and Atsushi Iwasaki. Approximately Stable Matchings with Budget Constraints. In *Proceedings of AAAI*, pages 242–248, 2018.
15 Yasushi Kawase and Atsushi Iwasaki. Approximately Stable Matchings with General Constraints. *CoRR*, abs/1907.04163, 2019. arXiv:1907.04163.
16 Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer-Verlag Berlin Heidelberg, 2004.
17 Anton J. Kleywegt and Jason D. Papastavrou. The dynamic and stochastic knapsack problem. *Operations research*, 46(1):17–35, 1998.
18 Dennis Komm. *An Introduction to Online Computation*. Springer, 2016.
19 Alberto Marchetti-Spaccamela and Carlo Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68(1):73–104, 1995.
20 Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems. In *Proceedings of WWW*, pages 1243–1244, 2008.


## A Relationship Among $m, \epsilon$ and $R$ in Algorithm 2

Here, we prove some relationships among $m, \epsilon$ and $R$ in Algorithm 2.

▶ **Lemma 18.** *Let $R \geq 3$, $m := \lfloor (R-3)/2 \rfloor$ and let $\epsilon > 0$ be a real such that $\log_{1+\epsilon}(1/\epsilon) = m$. Then, $m = \Theta(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ and $\epsilon = O(\log R/R)$*

**Proof.** By the definition of the base of natural logarithm $e$ and the monotonicity of $(1+1/x)^x$, we have $2 \leq (1+1/x)^x \leq e$ for any $x \geq 1$. As $\epsilon \leq 1$, we have

$$2^{\epsilon m} \leq (1+\epsilon)^{\frac{1}{\epsilon}\epsilon m} \leq e^{\epsilon m}.$$

By substituting $m = \log_{1+\epsilon}(1/\epsilon)$, we have $(1+\epsilon)^{\frac{1}{\epsilon}\epsilon m} = 1/\epsilon$. Hence, we get

$$\epsilon m \log 2 \leq \log \frac{1}{\epsilon} \leq \epsilon m. \tag{1}$$

This implies $m = \Theta(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$.

Next, we show that $\epsilon = O(\log R/R)$. By the inequalities (1), we have

$$\epsilon \leq \frac{\log \frac{1}{\epsilon}}{m \log 2} \leq \frac{\log\left(\frac{1}{\epsilon} \log \frac{1}{\epsilon}\right)}{m \log 2} \leq \frac{\log m}{m \log 2} = \frac{\log \lfloor (R-3)/2 \rfloor}{\lfloor (R-3)/2 \rfloor \log 2} = O\left(\frac{\log R}{R}\right).$$ ◀