Online Knapsack Problem with Removal Cost

Xin Han[1,*], Yasushi Kawase[2,**], and Kazuhisa Makino[2,**]

[1] Dalian University of Technology
hanxin@dlut.edu.cn
[2] University of Tokyo
{yasushi_kawase,makino}@mist.i.u-tokyo.ac.jp

Abstract. In this paper, we study the online knapsack problem with removal cost. The input is a sequence of items $u_1, u_2, \ldots, u_n$, each of which has a size and a value, where the value of each item is assumed to be equal to the size. Given the ith item u_i, we either put u_i into the knapsack or reject it with no cost. When u_i is put into the knapsack, some items in the knapsack are removed with removal cost if the sum of the size of u_i and the total size in the current knapsack exceeds the capacity of the knapsack. Here the removal cost means a cancellation charge or disposal fee. Our goal is to maximize the profit, i.e., the sum of the values of items in the last knapsack minus the total removal cost occurred.

In this paper, we consider two kinds of removal cost: unit and proportional cost. For both models, we provide their competitive ratios. Namely, we construct optimal online algorithms and prove that they are best possible.

1 Introduction

The knapsack problem is one of the most classical problems in combinatorial optimization and has a lot of applications in the real world [10]. The knapsack problem is that: given a set of items with values and sizes, we are asked to maximize the total value of selected items in the knapsack satisfying the capacity constraint.

In this paper, we study the online version of the knapsack problem with removal cost. Here, "online" means i) the information of the input (i.e., the items) is given gradually, i.e., after a decision is made on the current item, the next item is given; ii) the decisions we have made are irrevocable, i.e., once a decision has been made, it cannot be changed. Given the ith item u_i, we either accept u_i (i.e., put u_i into the knapsack) or reject it with no cost. When u_i is put into the knapsack, some items in the knapsack are removed with removal cost if the sum of the size of u_i and the total size in the current knapsack exceeds

[*] Partially supported by NSFC(11101065).

[**] Supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan and the Global COE "The Research and Training Center for New Development in Mathematics."

J. Gudmundsson, J. Mestre, and T. Viglas (Eds.): COCOON 2012, LNCS 7434, pp. 61–73, 2012.

1, i.e., the capacity of the knapsack. Here the removal cost means a cancellation charge or disposal fee. Our goal is to maximize the profit, i.e., the sum of the values of items in the last knapsack minus the total removal cost occurred.

1.1 Related Work

The online knapsack problem (under no removal condition) was first studied on average case analysis by Marchetti-Spaccamela and Vercellis [12]. They proposed a linear time approximation algorithm such that the expected difference between the optimal profit and the one obtained by the algorithm is $O(\log^{3/2} n)$ under the condition that the capacity of the knapsack grows proportionally to the number of items n. Lueker [11] improved the expected difference to $O(\log n)$ under a fairly general condition on the distribution.

Iwama and Taketomi [8] studied the online knapsack problem on worst case analysis. They obtained a $\frac{1+\sqrt{5}}{2} \approx 1.618$-competitive algorithm for the online knapsack when (1) the removable condition (without removal cost) is allowed and (2) the value of each item is equal to the size, and showed that this is best possible by providing a lower bound 1.618 for the case. We remark that the problem has unbounded competitive ratio, if at least one of the conditions (1) and (2) is not satisfied [8, 9]. For other models such as minimum knapsack problem and knapsack problem with limited cuts, refer to papers in [6, 7, 13].

The *removal cost* has introduced in the buyback problem [1–5]. In the problem, we observe a sequence of bids and decide whether to accept each bid at the moment it arrives, subject to constraints on accepted bids such as single item and matroid constraints. Decisions to reject bids are irrevocable, whereas decisions to accept bids may be canceled at a cost which is a fixed fraction of the bid value. Babaioff *et al.* [3] showed that the buyback problem with matroid constraint has $\left(1+2f+2\sqrt{f(1+f)}\right)$-competitive ratio, where $f > 0$ is a buyback factor. Ashwinkumar [1] extended their results and show that the buyback problem with the constraint of k matroid intersections has $k(1+f)(1+\sqrt{1-\frac{1}{k(1+f)}})^2$-competitive ratio.

1.2 Our Results

In this paper, we study the worst case analysis of the online knapsack problem with removal cost, when the value of each item is equal to the size. We considers two kinds of models of removal cost: the *proportional* and the *unit* cost model. In the proportional cost model, the removal cost of each item u_i is proportional to its value (and hence size), i.e., it is $f \cdot s(u_i)$, where $s(u_i)$ denotes the size of u_i and $f > 0$ is a fixed constant, called *buyback factor*. Therefore, we can view this model as the buyback problem with knapsack constraints. In the unit cost model, the removal cost of each item is a fixed constant $c > 0$, where we assume that every item has value at least c, since in many applications, the removal cost (i.e., cancellation charge) is not higher than its value. We remark that the problem has unbounded competitive ratio if no such assumption is satisfied (see Section 3).

We show that the proportional and unit cost models have competitive ratios $\lambda(f)$ and $\mu(c)$ in (1) and (2), respectively, where $\lambda(f)$ and $\mu(c)$ are given in Figure 1. Namely, we construct $\lambda(f)$- and $\mu(c)$-competitive algorithms for the models and prove that they are best possible.

$$\lambda(f) = \begin{cases} 2 & (1/2 \geq f > 0), \\ \frac{1+f+\sqrt{f^2+2f+5}}{2} & (f > 1/2). \end{cases} \tag{1}$$

$$\mu(c) = \begin{cases} \max\{\eta(k),\, \xi(k+1)\} & (1 - \sqrt{\frac{k+1}{k+2}} \leq c \leq 1 - \sqrt{\frac{k}{k+1}},\ k = 1, 2, \ldots), \\ \xi(1) & (1 - \frac{1}{\sqrt{2}} \leq c \leq 1/2), \\ 1/c & (c \geq 1/2), \end{cases} \tag{2}$$

where

$$\eta(k) = \frac{k(c+1) + \sqrt{k^2(1-c)^2 + 4k}}{2k(1-kc)}, \quad \xi(k) = \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4}{kc}}. \tag{3}$$

The main ideas of our algorithms for both models are: i) we may reject items (with no cost) many times, but in at most one round, we remove items which from the knapsack. ii) some items are removed from the knapsack, only when the total value in the resulting knapsack gets high enough to guarantee the optimal competitive ratio.

The rest of the paper is organized as follows. In the next section, we consider the proportional cost model, and in Section 3, we consider the unit cost model. Due to the space limitation, some of the proofs are omitted.

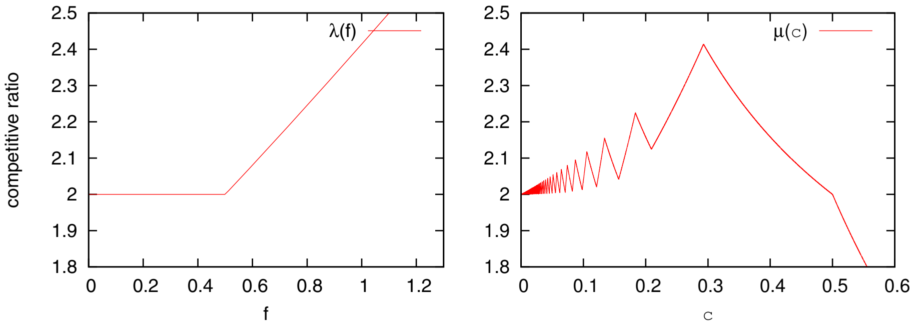

Fig. 1. The competitive ratios $\lambda(f)$ and $\mu(c)$ for the proportional and unit cost models

2 Proportional Cost Model

In this section, we consider the proportional cost model, where each item u_i has removal cost $f \cdot s(u_i)$ for some positive constant f. We first show that $\lambda(f)$ is a lower bound of the competitive ratio of the problem, and then propose a $\lambda(f)$-competitive algorithm, where $\lambda(f)$ is given in (1).

2.1 Lower Bound

In this subsection, we show a lower bound of the competitive ratio $\lambda(f)$ for the problem.

Theorem 1. *There exists no online algorithm with competitive ratio strictly less than $\rho = \lambda(f)$ for the knapsack problem with proportional removal cost, when the value of each item is equal to the size.*

Proof. According to the value of f, we separately consider the following two cases.

Case 1: $1/2 \geq f > 0$. Let A denote an online algorithm chosen arbitrarily. For a sufficiently small $\varepsilon\,(> 0)$, our adversary requests the sequence of items whose sizes are

$$\frac{1}{2}+\varepsilon,\ \frac{1}{2}+\frac{\varepsilon}{2},\ \ldots,\ \frac{1}{2}+\frac{\varepsilon}{\lceil 1/f\rceil+1}, \tag{4}$$

until A rejects some item in (4). If A rejects the item with size $\frac{1}{2}+\varepsilon$, then the adversary stops the input sequence. On the other hand, if it rejects the item with size $\frac{1}{2}+\frac{\varepsilon}{k}$ for some $k > 1$, then the adversary requests an item with size $\frac{1}{2}-\frac{\varepsilon}{k}$ and stops the input sequence.

We first note that algorithm A must take the first item, since otherwise the competitive ratio of A becomes infinite. After the first round, A always keeps exactly one item in the knapsack, since all the items in (4) have size larger than $\frac{1}{2}$ (i.e., a half of the knapsack capacity) and for any $j < k$ we have $(\frac{1}{2}+\frac{\varepsilon}{j})+(\frac{1}{2}-\frac{\varepsilon}{k})$ is larger than 1. This implies that A removes the old item from the knapsack to accept a new item. If A rejects $\frac{1}{2}+\frac{\varepsilon}{k}$ for some $k > 1$, the competitive ratio is at least $1/\left(\frac{1}{2}+\frac{\varepsilon}{k}\right)$, which approaches 2 as $\varepsilon \to 0$. Finally, if A rejects no item in (4), then its profit is

$$\frac{1}{2}+\frac{\varepsilon}{\lceil 1/f\rceil+1}-f\sum_{k=1}^{\lceil 1/f\rceil}\left(\frac{1}{2}+\frac{\varepsilon}{k}\right) \leq \frac{1}{2}-f\sum_{i=1}^{\lceil 1/f\rceil}\frac{1}{2} \leq 0 \tag{5}$$

while the optimal profit for the offline problem is $\frac{1}{2}+\varepsilon$, which completes the proof for $1/2 \geq f > 0$.

Case 2: $f > 1/2$. Let A denote an online algorithm chosen arbitrarily, and let $x = \frac{3+f-\sqrt{f^2+2f+5}}{2(1+f)}$. For a sufficiently small $\varepsilon\,(> 0)$, our adversary requests the following sequence of items

$$x,\ 1-x+\varepsilon,\ 1-x, \tag{6}$$

until A rejects some item in (6), and if A rejects the item then the adversary immediately stops the input sequence.

Note that A must accept the first item x, since otherwise the competitive ratio becomes infinite. If A rejects the second item, then the competitive ratio is at least

$$\frac{1-x+\varepsilon}{x} \geq \frac{1-x}{x} = \lambda(f). \tag{7}$$

If A takes the second item $1-x+\varepsilon$ (and removes the first item),the competitive ratio is at least $\frac{1}{1-x+\varepsilon-f\cdot x}$, which approaches to $\lambda(f)\,(=\frac{1}{1-x-f\cdot x})$ as $\varepsilon \to 0$, which completes the proof for $f > 1/2$. □

2.2 Upper Bound

In this subsection, we propose a $\lambda(f)$-competitive algorithm. Note that the total profit becomes small (even negative), if we remove items from the knapsack many times. Intuitively, our algorithm accepts the item if the knapsack has room to put it. If we can make the profit sufficiently high by accepting the item and removing some items from the current knapsack, then our algorithm follows this, and after this iteration, it rejects all the items. Otherwise, we simply rejects the item.

Let $\rho = \lambda(f)$, and let u_i be the item given in the ith round. Define by B_{i-1} the set of items in the knapsack at the beginning of ith round, and by $s(B_{i-1})$ the total size in B_{i-1}.

```
Algorithm 1.
  1. if  $s(u_i) + s(B_{i-1}) \leq 1$  then  $B_i \leftarrow B_{i-1} \cup \{u_i\}$ and if $s(B_i) \geq 1/\rho$
     then STOP
  2. else if  $\exists B'_i \subseteq B_{i-1}$ s.t. $\frac{1}{\rho} + f \cdot (s(B_{i-1}) - s(B'_i)) < s(B'_i) + s(u_i) \leq 1$
     then  $B_i \leftarrow B'_i \cup \{u_i\}$ and STOP
  3. else  $B_i \leftarrow B_{i-1}$
```

Here STOP denotes that the algorithm rejects the items after this round.

Lemma 2. *If $s(u_i) + s(B_{i-1}) > 1$ and some $B'_i \subseteq B_{i-1}$ satisfies $\rho \cdot s(B_{i-1}) < s(B'_i) + s(u_i) \leq 1$, then the second line is executed in the ith round.*

Proof. Since $s(u_i) + s(B_{i-1}) > 1$ and $\rho \cdot s(B_{i-1}) < s(B'_i) + s(u_i)$, we obtain

$$\begin{aligned}
\frac{1}{\rho} + f \cdot (s(B_{i-1}) - s(B'_i)) &< \frac{s(u_i) + s(B_{i-1})}{\rho} + f \cdot (s(B_{i-1}) - s(B'_i)) \\
&< \frac{s(u_i)}{\rho} + \frac{s(B'_i) + s(u_i)}{\rho^2} + f \cdot \frac{s(B'_i) + s(u_i)}{\rho} - fs(B'_i) \\
&= \frac{1 + f\rho - f\rho^2}{\rho^2} s(B'_i) + \frac{1 + f\rho + \rho}{\rho^2} s(u_i).
\end{aligned} \tag{8}$$

As $\rho^2 \geq 1 + f\rho + \rho$ by the definition of ρ, we have

$$\frac{1 + f\rho - f\rho^2}{\rho^2} \leq \frac{1 + f\rho - f\rho^2}{1 + f\rho + \rho} < 1 \quad \text{and} \quad \frac{1 + f\rho + \rho}{\rho^2} \leq 1. \tag{9}$$

□

Let OPT denote an optimal solution for the offline problem whose input sequence is $u_1, \ldots, u_i$.

Lemma 3. *If $s(B_i) < 1/\rho$ then we have $|\mathrm{OPT} \setminus B_i| \leq 1$.*

Proof. B_i contains all the items smaller than $1/2$, since $s(B_i) < 1/\rho \leq 1/2$. Any item $u \in \mathrm{OPT} \setminus B_i$ has size greater than $1 - 1/\rho \geq 1/2$. Therefore, $|\mathrm{OPT} \setminus B_i| \leq 1$ holds by $s(\mathrm{OPT}) \leq 1$. □

Theorem 4. *The online algorithm given in this section is $\lambda(f)$-competitive.*

Proof. Suppose that the second line is executed in round k. Then it holds that $\frac{1}{\rho} + f \cdot (s(B_{k-1}) - s(B'_k)) < s(B'_k) + s(u_k) = s(B_k)$ holds. Since $s(B_i) = s(B_k)$ holds for all $i \geq k$, we have

$$\frac{s(\mathrm{OPT})}{s(B_i) - f \cdot (s(B_{k-1}) - s(B'_k))} \leq \frac{1}{s(B_k) - f \cdot (s(B_{k-1}) - s(B'_k))} < \rho \, (= \lambda(f)). \tag{10}$$

We next assume that the second line has never been executed. If $s(B_i) \geq 1/\rho$, we have the competitive ratio $s(\mathrm{OPT})/s(B_i) \leq 1/s(B_i) \leq \rho$. On the other hand, if $s(B_i) < 1/\rho$, $|\mathrm{OPT} \setminus B_i| = 0$ or 1 holds by Lemma 3, If $|\mathrm{OPT} \setminus B_i| = 0$, we obtain the competitive ratio 1. Otherwise (i.e., $\mathrm{OPT} \setminus B_i = \{u_k\}$ for some k), Lemma 2 implies that $\rho \cdot s(B_{k-1}) \geq s(B'_k) + s(u_k)$ for $B'_k = \mathrm{OPT} \cap B_{k-1}$ Therefore we obtain

$$\begin{aligned} \frac{s(\mathrm{OPT})}{s(B_i)} &\leq \frac{s(B'_k) + s(u_k) + s(B_i \setminus B_{k-1})}{s(B_{k-1}) + s(B_i \setminus B_{k-1})} \\ &\leq \max\left\{ \frac{s(B'_k) + s(u_k)}{s(B_{k-1})}, \frac{s(B_i \setminus B_{k-1})}{s(B_i \setminus B_{k-1})} \right\} \leq \rho \, (= \lambda(f)). \end{aligned} \tag{11}$$

□

Before concluding this section, we remark that the condition in the second line can be checked in $O(|B_{i-1}| + 2^{\rho^2})$ time.

3 Unit Cost Model

In this section, we consider the unit cost model, where it costs us a fixed constant $c > 0$ to remove each item from the knapsack. Recall that every item has size at least c. In this section, we show that the knapsack problem with unit cost is $\mu(c)$-competitive, where $\mu(c)$ is defined in (2). We note that $\mu(c)$ attains the maximum $1+\sqrt{2}$ when $c = 1 - 1/\sqrt{2}$.

Remark: If items are allowed to have size smaller than c, the problem becomes unbounded competitive. To see this, for a positive number r, let ε denote a positive number such that $\varepsilon < 1/(\lceil 1/c \rceil \cdot r)$. For an online algorithm A chosen arbitrarily, our adversary keeps requesting the items with size ε, until A accepts $\lceil 1/c \rceil$ items or rejects $r \cdot \lceil 1/c \rceil$ items. If A rejects $r \cdot \lceil 1/c \rceil$ items (before accepting $\lceil 1/c \rceil$ items), the adversary stops the input sequence; otherwise, it requests an item with size 1 and stops the input sequence. In the former case, the competitive ratio is at least $\frac{r\lceil 1/c \rceil \varepsilon}{\lceil 1/c \rceil \varepsilon} = r$. In the latter case, the competitive ratio becomes $\frac{1}{\lceil 1/c \rceil \cdot \varepsilon} > r$ if A rejects the last item (with size 1). Otherwise, A removes the $\lceil 1/c \rceil$ items to take the last item. This implies that the profit is $1 - \lceil 1/c \rceil \cdot c \leq 0$. Therefore, without the assumption, no online algorithm attains a bounded competitive ratio.

3.1 The Case $c \geq 1/2$

We first consider the case where $c \geq 1/2$. In this case, it is not difficult to see that the problem is $1/c\,(= \mu(c))$-competitive.

Theorem 5. *If the unit removal cost c of the knapsack problem is at least* $1/2$*, then there exists no online algorithm with competitive ratio strictly less than* $1/c$ *for the problem when the value of each item is equal to the size.*

Proof. For an online algorithm A chosen arbitrarily, our adversary first requests an item with size c. If A does not accept it, the adversary stop the input sequence. Otherwise, it next request an item with size 1 and stop the input sequence. It is clear that A must take the first item, since otherwise the competitive ratio becomes infinite. If A rejects the second item, then we have the competitive ratio $1/c$. Otherwise (i.e., A accepts the second item by removing the first item), the competitive ratio is $1/(1-c) \geq 1/c$, since $c \geq 1/2$. □

Theorem 6. *There exists a* $1/c$*-competitive algorithm for the knapsack problem with unit removal cost, when the value of each item is equal to the size.*

Proof. Consider an online algorithm which takes the first item u_1 and rejects the remaining items. Since $s(u_1) \geq c$ and the optimal value of the offline problem is at most 1, the competitive ratio is at most $1/c$. □

3.2 The Case $c < 1/2$

In this section we consider the case in which $c < 1/2$.

3.2.1 Lower Bound

For $0 < c < 1/2$, we show that $\mu(c)$ is a lower bound of the competitive ratio for the problem by starting with several propositions needed later.

Proposition 7. *For any positive integer k, we have*

$$\frac{1}{2k+4} < 1 - \sqrt{\frac{k+1}{k+2}}, \quad 1 - \sqrt{\frac{k}{k+1}} < \frac{1}{2k+1}. \tag{12}$$

Definition 8. *We define x_k and y_k as follows:*

$$x_k = \frac{k+2-kc-\sqrt{k^2(1-c)^2+4k}}{2}, \quad y_k = \frac{kc+\sqrt{k^2c^2+4kc}}{2}. \tag{13}$$

Proposition 9. *$\eta(k)$ and $\xi(k)$ in (3) satisfy the following equalities.*

$$\eta(k) = \frac{1}{1-x_k-kc} = \frac{1-x_k}{kx_k} = \frac{k(c+1)+\sqrt{k^2(1-c)^2+4k}}{2k(1-kc)}, \tag{14}$$

$$\xi(k) = \frac{1}{y_k-kc} = \frac{y_k}{kc} = \frac{1}{2} + \frac{1}{2}\sqrt{1+\frac{4}{kc}}. \tag{15}$$

We provide two kinds of adversaries.

Theorem 10. *Assume that removal cost c satisfies $1-\sqrt{\frac{k+1}{k+2}} \le c \le 1-\sqrt{\frac{k}{k+1}}$ for a positive integer k. Then there exists no online algorithm with competitive ratio strictly less than $\eta(k)$ for the knapsack problem with unit removal cost, when the value of each item is equal to the size.*

Proof. Let $x_k = \frac{k+2-kc-\sqrt{k^2(1-c)^2+4k}}{2}$. For an online algorithm A chosen arbitrarily, our adversary keeps requesting the items with size x_k until A accepts k items or rejects $\lceil 1/x_k \rceil$ items. If A rejects $\lceil 1/x_k \rceil$ items before accepting k items, the adversary stops the input sequence (1). Otherwise (i.e., A accepts k items), then the adversary next requests an item with size $1-x_k+\varepsilon$ where ε is a sufficiently small positive number; if A rejects it, the adversary stops the input sequence (2), and otherwise, the adversary next requests an item with size $1-x_k$ and stops the input sequence (3). Note that all the items have size at least c, since $1-\sqrt{\frac{k+1}{k+2}} \le c \le 1-\sqrt{\frac{k}{k+1}}$ implies $x_k \ge c$ and $1-x_k \ge c$.

In the case of (1), we have the competitive ratio at least $\frac{1-x_k}{(k-1)x_k} > \frac{1-x_k}{kx_k} = \eta(k)$, where the last equality follows from Proposition 9. In the case of (2), the competitive ratio is at least $\frac{1-x_k+\varepsilon}{kx_k} > \frac{1-x_k}{kx_k} = \eta(k)$ by Proposition 9. Finally, in the case of (3), the competitive ratio is at least $\frac{1}{1-x_k+\varepsilon-kc}$. Proposition 9 implies that this approaches $\eta(k)$ $(= \frac{1}{1-x_k-kc})$ as $(\varepsilon \to 0)$. □

Theorem 11. *Assume that removal cost c satisfies $1-\sqrt{\frac{k}{k+1}} \le c < \frac{1}{2k}$ for a positive integer k. Then there exists no online algorithm with competitive ratio strictly less than $\xi(k)$ for the knapsack problem with unit removal cost, when the value of each item is equal to the size.*

Proof. Let A denote an online algorithm chosen arbitrarily. Then our adversary keeps requesting the items with size c until A accepts k items or rejects $\lceil 1/c \rceil$

items. If A rejects $\lceil 1/c \rceil$ items before accepting k items, the adversary stops the input sequence (1). Otherwise (i.e., A accepts k items), the adversary requests an item with size $y_k = \frac{kc+\sqrt{k^2c^2+4kc}}{2}$ which is at least $1-c > c$, since $1-\sqrt{\frac{k}{k+1}} \leq c < \frac{1}{2k}$; if A rejects it, the adversary stops the input sequence (2), and otherwise, the adversary requests an item with size $1-c$ and stops the input sequence (3).

In the case of (1), the competitive ratio is at least $\frac{1-c}{(k-1)c} \geq \frac{1}{kc} \geq \frac{y_k}{kc} = \xi(k)$, where the last equality follows from Proposition 9. In the case of (2), the competitive ratio is $\frac{y_k}{kc} = \xi(k)$ by Proposition 9. Finally, in the case of (3), the competitive ratio is at least $\frac{1}{y_k - kc} = \xi(k)$, which again follows from Proposition 9. □

By Theorems 10 and 11, it holds that $\mu(c)$ is a lower bound of the competitive ratio for $0 < c < 1/2$.

3.2.2 Upper Bound

In this subsection, we show $\mu(c)$ is also an upper bound for the competitive ratio of the problem when $0 < c < 1/2$. We start with several propositions needed later.

Proposition 12. *For a positive integer k, let c satisfy $0 < c \leq 1-\sqrt{\frac{k}{k+1}}$. Then we have*

$$\max\{\eta(k),\, \xi(k+1)\} \geq 2. \tag{16}$$

Proposition 13. *For a positive integer k, let c satisfy $1-\sqrt{\frac{k+1}{k+2}} \leq c \leq 1-\sqrt{\frac{k}{k+1}}$. Then we have*

$$\max\left\{\max_{\alpha\in\{1,2,\dots,k\}} \eta(\alpha),\, \xi(k+1)\right\} = \max\{\eta(k),\, \xi(k+1)\} = \mu(c). \tag{17}$$

Proposition 14. *For a positive integer k, let c satisfy $1-\sqrt{\frac{k+1}{k+2}} \leq c \leq 1-\sqrt{\frac{k}{k+1}}$. Then for any positive integer $\alpha \leq k$ and real $x \in (0, 1-\alpha c)$, it holds that*

$$\min\left\{\frac{1}{1-x-\alpha c}, \frac{1-x}{\alpha x}\right\} \leq \eta(\alpha) \leq \mu(c). \tag{18}$$

Proof. Since $\frac{1}{1-x-\alpha c}$ and $\frac{1-x}{\alpha x}$ are respectively monotone increasing and decreasing in x, the first inequality holds by Proposition 9. The second inequality is obtained by Proposition 13. □

Proposition 15. *For a positive integer k, let c satisfy $1-\sqrt{\frac{k+1}{k+2}} \leq c \leq 1-\sqrt{\frac{k}{k+1}}$. Then for any real $y \in ((k+1)c, 1]$, we have*

$$\min\left\{\frac{1}{y-(k+1)c}, \frac{y}{(k+1)c}\right\} \leq \xi(k+1) \leq \mu(c). \tag{19}$$

Proof. Since $\frac{1}{y-(k+1)c}$ and $\frac{y}{(k+1)c}$ are respectively monotone decreasing and increasing in y, the first inequality holds by Proposition 9. The second inequality follows from the definition of $\mu(c)$. □

We are now ready to prove that $\mu(c)$ is an upper bound for the competitive ratio. According to the size of c, we make use of two algorithms described below.

Theorem 16. *If* $1-\frac{1}{\sqrt{2}} \leq c \leq \frac{1}{2}$, *there exists an online algorithm with competitive ratio* $\mu(c)$ *for the knapsack problem with unit removal cost, when the value of each item is equal to the size.*

Proof. We consider the following algorithm, where B_{i-1} denotes the set of items in the knapsack at the beginning of the ith round, and and $s(B_{i-1})$ denotes the total size in B_{i-1}. Let u_i be the item given in the ith round.

```
Algorithm 2.
 1. if $s(B_{i-1}) + s(u_i) \leq 1$ then $B_i \leftarrow B_{i-1} \cup \{u_i\}$
 2. else if $|B_{i-1}| = 1$ and $s(u_i) \geq \frac{c+\sqrt{c^2+4c}}{2}$ then $B_i \leftarrow \{u_i\}$ and STOP
 3. else $B_i \leftarrow B_{i-1}$
```

Here STOP denotes that the algorithm rejects the items after this round.

Let OPT denote an optimal solution for the offline problem whose input sequence is $u_1, \ldots, u_i$. If the algorithm stops at the second line, the competitive ratio is at most $1/\left(\frac{c+\sqrt{c^2+4c}}{2} - c\right) = \frac{c+\sqrt{c^2+4c}}{2c} = \mu(c)$, since $s(\mathrm{OPT}) \leq 1$. Assume that the algorithm has never stopped at the second line and $|B_i| = 1$. Then if $s(B_i) \geq 1/2$, the competitive ratio is at most $\frac{1}{1/2} = 2 \leq \mu(c)$. Otherwise, the item in B_i has size smaller than $1/2$, while the item u_j with $j < i$ and $u_j \notin B_i$ has size at least $1/2$. This implies that $|\mathrm{OPT}| = 1$ and the competitive ratio is smaller than $\mu(c)$, since $s(B_i) \geq c$ and $s(\mathrm{OPT}) < \frac{c+\sqrt{c^2+4c}}{2}$. If the algorithm has never stopped at the second line and $|B_i| > 1$, the competitive ratio is at most $\frac{1}{2c} < \mu(c)$, since $c \geq 1 - 1/\sqrt{2} > 1/6$ implies $c + \sqrt{c^2+4c} > 1$. □

Theorem 17. *If* $1-\sqrt{\frac{k+1}{k+2}} \leq c \leq 1-\sqrt{\frac{k}{k+1}}$, *there exists an online algorithm with competitive ratio* $\mu(c)$ *for the knapsack problem with unit removal cost, when the value of each item is equal to the size.*

Proof. We show that the following algorithm satisfies the desired property. In the algorithm, let $B_{i-1} = \{b_1, b_2, \ldots, b_m\}$ be the set of items in the knapsack at the beginning of the ith round, such that $s(b_1) \geq s(b_2) \geq \cdots \geq s(b_m)$. Let u_i be the item given in the ith round.

```
Algorithm 3.
1. if $s(B_{i-1}) + s(u_i) \le 1$ then $B_i \leftarrow B_{i-1} \cup \{u_i\}$
2. else
3.     $B'_{i-1} \leftarrow \emptyset$
4.     for $j = 1$ to $m$ if $s(B'_{i-1}) + s(b_j) \le 1 - s(u_i)$ then $B'_{i-1} \leftarrow B'_{i-1} \cup \{b_j\}$
5.      if $s(B'_{i-1}) + s(u_i) - |B_{i-1} \setminus B'_{i-1}|c \ge 1/\mu(c)$
   then $B_i \leftarrow B'_{i-1} \cup \{u_i\}$ and STOP
6.      else $B_i \leftarrow B_{i-1}$
```

Here STOP denotes that the algorithm rejects the items after this round.

Let OPT denote an optimal solution for the offline problem whose input sequence is $u_1, \dots, u_i$. If the algorithm stops at the fifth line in round $l \le i$, $s(B_i) = s(B_l) = s(B'_{l-1}) + s(u_l)$ and the profit of the algorithm is $s(B'_{l-1}) + s(u_l) - |B_{l-1} \setminus B'_{l-1}|c$. Therefore, the competitive ratio is at most $\frac{1}{s(B'_{l-1}) + s(u_l) - |B_{l-1} \setminus B'_{l-1}|c} \le \mu(c)$, since $s(\mathrm{OPT}) \le 1$. Otherwise, the algorithm has never removed old items from the knapsack. If $s(B_i) \ge 1/2$, then the competitive ratio is at most $\frac{1}{1/2} = 2 \le \mu(c)$. On the other hand, if $s(B_i) < 1/2$, then any item in B_i has size at most 1/2. while any item in $OPT \setminus B_i$ has size larger than 1/2. This implies $|\mathrm{OPT} \setminus B_i| \le 1$ by $s(\mathrm{OPT}) \le 1$. If $|\mathrm{OPT} \setminus B_i| = 0$, then we have $\mathrm{OPT} = B_i$, which implies that the competitive ratio is 1. Thus we assume that $|\mathrm{OPT} \setminus B_i| = 1$. For the cardinality of B_i, we have $|B_i| \le k + 1$, since any $b \in B_i$ satisfies $s(b) \ge c \ge 1 - \sqrt{\frac{k+1}{k+2}} \ge \frac{1}{2k+4}$, where the last inequality follows from Proposition 7. Since the algorithm has never removed items, $|B_l| \le k + 1$ also holds for each l with $l \le i$. Let

$$\{u_l\} = \mathrm{OPT} \setminus B_i,\ \alpha = |B_{l-1} \setminus B'_{l-1}|,\ x = 1 - (s(u_l) + s(B'_{l-1})). \tag{20}$$

Since $B_{l-1} \setminus B'_{l-1} \ne \emptyset$, we have

$$\alpha > 0 \ \text{ and } \ x < \sqrt{\frac{k}{k+1}} < 1 - \alpha c. \tag{21}$$

Since $s(B_i) = s(B_{l-1}) + s(B_i \setminus B_{l-1})$ and $s(\mathrm{OPT}) \le s(u_l) + s(B_{l-1} \cap \mathrm{OPT}) + s(B_i \setminus B_{l-1})$, the competitive ratio is at most

$$\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT}) + s(B_i \setminus B_{l-1})}{s(B_{l-1}) + s(B_i \setminus B_{l-1})} \le \max\left\{\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT})}{s(B_{l-1})}, 1\right\}.$$

We claim that $\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT})}{s(B_{l-1})} \le \mu(c)$.

Let $B_l = \{b_1, b_2, \dots, b_m\}$ satisfy $s(b_1) \ge s(b_2) \ge \cdots \ge s(b_m)$. To see this claim, we separately consider the following two cases:

Case 1. Consider the case in which there exists $b_j \in B'_{l-1}$ such that $b_h \notin B'_{l-1}$ holds for some $h > j$. Let us take b_j as the largest such item, i.e., $b_j \in B'_{l-1}$ and $b_g \notin B'_{l-1}$ for all $g\,(< j)$.

In this case, we obtain the following inequalities:

$$\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT})}{s(B_{l-1})} \le \frac{s(b_h) + 1 - x}{s(b_h) + \alpha x} \le \max\left\{1, \frac{1-x}{\alpha x}\right\}. \tag{22}$$

Here the numerator and denominator in the left hand side of (22) respectively satisfy $s(u_l) + s(B_{l-1} \cap \mathrm{OPT}) \leq 1 < s(b_h) + s(u_l) + s(B'_{l-1}) = s(b_h) + 1 - x$ and $s(B_{l-1}) = s(B'_{l-1}) + s(B_{l-1} \setminus B'_{l-1}) \geq s(b_h) + \alpha x$, since $b_h \notin B'_{l-1}$ and $s(b) > x$ holds for any $b \in B_{l-1} \setminus B'_{l-1}$. Finally, we show $\frac{1-x}{\alpha x} \leq \mu(c)$, which completes the claim.

Since the algorithm has not stopped at the fifth line and $1-x-\alpha c > 0$ by (21), we have $\frac{1}{1-x-\alpha c} = \frac{1}{s(B'_{l-1})+s(u_l)-\alpha c} > \mu(c)$. Note that $\alpha \leq |B_{l-1} \setminus \{b_h\}| \leq k$, since $|B_{l-1}| \leq k+1$. Therefore, we obtain $\frac{1-x}{\alpha x} \leq \mu(c)$ by Proposition 14.

Case 2. We next consider the case in which $b_j \in B'_{l-1}$ implies $b_h \in B'_{l-1}$ for all $h\,(> j)$, i.e., B'_{l-1} consists of the $|B'_{l-1}|$ smallest items of B_{l-1}. Then we have $s(b) > 1 - s(u_l)$ for any $b \in B_{l-1} \setminus B'_{l-1}$. This implies $B_{l-1} \cap \mathrm{OPT} \subseteq B'_{l-1}$, and $s(B_{l-1} \setminus B'_{l-1}) > \alpha x$ holds by (20).

If $\alpha \leq k$, thus, the competitive ratio is at most

$$\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT})}{s(B_{l-1})} \leq \frac{s(u_l) + s(B'_{l-1})}{s(B_{l-1} \setminus B'_{l-1})} \leq \frac{1-x}{\alpha x} \leq \mu(c), \tag{23}$$

where the last inequality follows from a similar argument to **Case 1**. On the other hand, if $\alpha = k+1$, let $y = s(u_l) + s(B'_{l-1})$. Then we have

$$\frac{s(u_l) + s(B_{l-1} \cap \mathrm{OPT})}{s(B_{l-1})} \leq \frac{y}{(k+1)c}, \tag{24}$$

where the inequality follows from the fact that $s(u_l) + s(B_{l-1} \cap \mathrm{OPT}) \leq s(u_l) + s(B'_{l-1}) = y$ and $s(B_{l-1}) \geq s(B'_{l-1}) \geq (k+1)c$, since $B_{l-1} \cap \mathrm{OPT} \subseteq B'_{l-1}$ and any item has size at least c. Finally, since $y > (k+1)c$ and the algorithm has not stopped at the fifth line, it holds that $\frac{1}{y-(k+1)c} = \frac{1}{s(B'_{l-1})+s(u_l)-(k+1)c} > \mu(c)$. This together with Proposition 15 implies $\frac{y}{(k+1)c} \leq \mu(c)$. □

References

1. Ashwinkumar, B.V.: Buyback Problem - Approximate Matroid Intersection with Cancellation Costs. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6755, pp. 379–390. Springer, Heidelberg (2011)
2. Ashwinkumar, B.V., Kleinberg, R.: Randomized Online Algorithms for the Buyback Problem. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 529–536. Springer, Heidelberg (2009)
3. Babaioff, M., Hartline, J.D., Kleinberg, R.D.: Selling ad campaigns: Online algorithms with cancellations. In: ACM Conference on Electronic Commerce, pp. 61–70 (2009)
4. Biyalogorsky, E., Carmon, Z., Fruchter, G.E., Gerstner, E.: Research note: Overselling with opportunistic cancellations. Marketing Science 18(4), 605–610 (1999)
5. Constantin, F., Feldman, J., Muthukrishnan, S., Pál, M.: An online mechanism for ad slot reservations with cancellations. In: SODA, pp. 1265–1274 (2009)
6. Han, X., Makino, K.: Online Minimization Knapsack Problem. In: Bampis, E., Jansen, K. (eds.) WAOA 2009. LNCS, vol. 5893, pp. 182–193. Springer, Heidelberg (2010)

7. Han, X., Makino, K.: Online removable knapsack with limited cuts. Theoretical Computer Science 411, 3956–3964 (2010)
8. Iwama, K., Taketomi, S.: Removable online knapsack problems. LNCS, pp. 293–305 (2002)
9. Iwama, K., Zhang, G.: Optimal Resource Augmentations for Online Knapsack. In: Charikar, M., Jansen, K., Reingold, O., Rolim, J.D.P. (eds.) RANDOM 2007 and APPROX 2007. LNCS, vol. 4627, pp. 180–188. Springer, Heidelberg (2007)
10. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer (2004)
11. Lueker, G.S.: Average-case analysis of off-line and on-line knapsack problems. Journal of Algorithms 29(2), 277–305 (1998)
12. Marchetti-Spaccamela, A., Vercellis, C.: Stochastic on-line knapsack problems. Mathematical Programming 68, 73–104 (1995)
13. Noga, J., Sarbua, V.: An online partially fractional knapsack problem. In: ISPAN, pp. 108–112 (2005)