Vol.2013-AL-145 No.8
2013/11/6

# Randomized Algorithms for Online Knapsack Problems

Xin Han*, Yasushi Kawase†, and Kazuhisa Makino ‡

**Abstract**

In this paper, we study online knapsack problems. The input is a sequence of items $e_1, e_2, \ldots, e_n$, each of which has a size and a value. Given the $i$th item $e_i$, we either put $e_i$ into the knapsack or reject it. In the removable setting, when $e_i$ is put into the knapsack, some items in the knapsack are removed with no cost if the sum of the size of $e_i$ and the total size in the current knapsack exceeds the capacity of the knapsack. Our goal is to maximize the profit, i.e., the sum of the values of items in the last knapsack.

We present a simple randomized 2-competitive algorithm for the unweighted non-removable case and show that it is the best possible, where knapsack problem is called unweighted if the value of each item is equal to its size. For the removable case, we propose a randomized 2-competitive algorithm despite there is no constant competitive deterministic algorithm. We also provide a lower bound $1+1/e \approx 1.368$ for the competitive ratio. For the unweighted removable case, we propose a 10/7-competitive algorithm and provide a lower bound 1.25 for the competitive ratio.

## 1 Introduction

The knapsack problem is one of the most fundamental problems in combinatorial optimization and has a lot of applications in the real world [12]. The knapsack problem is that: given a set of items $e_i$ with values $v(e_i)$ and sizes $s(e_i)$, we are asked to maximize the total value of selected items in the knapsack satisfying the capacity constraint. Throughout this paper, we assume that the capacity of knapsack is 1.

In this paper, we study the online version of the knapsack problem. Here, "online" means that 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 $i$th item $e_i$, which has a value $v(e_i)$ and a size $s(e_i)$, we either accept $e_i$ (i.e., put $e_i$ into the knapsack) or reject it. In the removable setting, when $e_i$ is put into the knapsack, some items in the knapsack are removed with no cost if the sum of the sizes of $e_i$ and the total size in the current knapsack exceeds 1 (i.e., the capacity of the knapsack). Our goal is to maximize the profit, i.e., the sum of the values of items in the last knapsack.

**Related works** It is well known that offline knapsack problem is NP-hard but admits an FPTAS. Ito *et al.* [9] presented a constant-time randomized approximation algorithm by using weighted sampling.

An online knapsack problem was first studied on average case analysis by Marchetti-Spaccamela and Vercellis [14]. They proposed a linear-time algorithm with $O(\log^{3/2} n)$ expected competitive difference, under the condition that the capacity of the knapsack grows proportionally to the number of items $n$. Lueker [13] improved the expected competitive difference to $O(\log n)$ under a fairly general condition on the distribution.

On the worst case analysis, Marchetti-Spaccamela and Vercellis [14] showed that general online knapsack problem has no constant (deterministic) competitive ratio. Buchbinder and Naor [4] presented an $O(\log(U/L))$-competitive algorithm based on a general online primal-dual

*Software School, Dalian University of Technology
†University of Tokyo
‡Kyoto University

framework when the density of every element is in a known range $[L, U]$, and each size is assumed to be much smaller than the capacity of the knapsack. They also showed an $\Omega(\log(U/L))$ lower bound on the competitive ratio for the case. Zhou *et al.* [16] showed $\Omega(\log(U/L))$ is also lower bound for the randomized case, which implies that general online knapsack problem has no constant randomized competitive ratio.

Iwama and Taketomi [10] studied the *removable* online knapsack problem. They obtained a $(1+\sqrt{5})/2 \approx 1.618$-competitive algorithm for the unweighted online knapsack, where knapsack problem is called *unweighted* if the value of each item is equal to its size, and showed that this is the best possible by providing a lower bound $(1+\sqrt{5})/2$ for the case. We remark that the problem has unbounded competitive ratio, if at least one of the removal and unweighted conditions is not satisfied [10,11]. For the randomized competitive ratio of the general removable online knapsack problem, Babaioff *et al.* [2] showed a lower bound $5/4$.

Removable online knapsack problem with cancellation cost is studied in [1,2,7]. When the cancellation cost is proportional, i.e., it is $f$ times the total value of removed items, Babaioff *et al.* [1,2] showed that if each item has size at most $\gamma$, where $0 < \gamma < 1/2$, then the competitive ratio is at most $1 + 2f + 2\sqrt{f(1+f)}$ with respect to the optimal solution for the knapsack problem with capacity $(1-2\gamma)$. They also proposed a randomized $3(1 + 2f + 2\sqrt{f(1+f)})$-competitive algorithm for this problem. Han *et al.* [7] showed that unweighted version of this problem is $\max\{2, \frac{1+f+\sqrt{f^2+2f+5}}{2}\}$-competitive.

For the other models such as knapsack secretary problem, stochastic knapsack problem and minimization knapsack problem, refer to papers in [3,5,6,7,8].

### Our results

In this paper, we study the worst case analysis of randomized algorithms for online knapsack problem against an oblivious adversary.

We first provide a randomized 2-competitive algorithm for the unweighted non-removable online knapsack problem, and show that it is the best possible.

For the unweighted removable case, we propose a randomized 10/7-competitive algorithm. Our algorithm divides all the items into three groups, *small*, *medium* and *large*. If a large item comes, our algorithm chooses it and cancels all the items in the knapsack. Otherwise the algorithm first handles medium items, then apply a greedy algorithm for the small items. For medium items, it randomly selects the one among two deterministic subroutines. We also show that there exists no randomized online algorithm with competitive ratio less than 5/4 for the unweighted removable case.

For the general removable case, we present a simple randomized 2-competitive algorithm, which is an extension of famous 2-approximation greedy algorithm for offline knapsack problem. As a lower bound, we show that there exists no randomized online algorithm with competitive ratio less than $1 + 1/e$ for the general removable online knapsack problem.

We summarize the current status on competitive ratios for the online knapsack problem in Table 1, where our results are written in bold letters.

The rest of paper is organized as follows. In Section 2, we provide competitive ratio for the unweighted non-removable cases. In Sections 3 and 4, we consider the unweighted and general removable cases, respectively.

## 2 Unweighted Non-Removable Online Knapsack Problem

In this section we study the non-removable version of the unweighted online knapsack problem. We show that the problem is randomized 2-competitive.

In order to show the upper bound, we construct the following algorithm called TwoBins. Algorithm TwoBins virtually keeps two bins, and puts items into either of the bins if possible. The algorithm outputs the items contained in the one of the two bins, which is randomly chosen in advance.

Table 1: The current status on competitive ratios for online knapsack problems, where our results are written in bold letters.

| | | unweighted | | general | |
|---|---|---|---|---|---|
| | | lower bound | upper bound | lower bound | upper bound |
| non-removable | deterministic | $\infty$ [10] | | $\infty$ [14] | |
| | randomized | **2** | | $\infty$ [16] | |
| removable | deterministic | $\frac{1+\sqrt{5}}{2}$ [10] | | $\infty$ [11] | |
| | randomized | **5/4** | **10/7** | $\mathbf{1+\frac{1}{e}}$ | **2** |

Let $e_i$ be the item given in the $i$th round. Define by $B_i$ the set of selected items at the end of the $i$th round by Algorithm TwoBins. For $r = 1, 2$, define by $B_i^r$ the set of selected items at the end of the $i$th round in bin $r$. Then our algorithm TwoBins is represented as follows.

Algorithm TwoBins

$B_0,\ B_0^1,\ B_0^2 := \emptyset$
choose $r$ uniformly at random from $\{1, 2\}$
**for each** item $e_i$ in order of arrival **do**
  **if** $s(B_i^1) + s(e_i) \leq 1$ **then** $B_i^1 := B_{i-1}^1 \cup \{e_i\}$, $B_i^2 := B_{i-1}^2$
  **else if** $s(B_i^2) + s(e_i) \leq 1$ **then** $B_i^1 := B_{i-1}^1$, $B_i^2 := B_{i-1}^2 \cup \{e_i\}$
  **else** $B_i^1 := B_{i-1}^1$, $B_i^2 := B_{i-1}^2$
  **if** $r = 1$ **then** $B_i := B_i^1$
  **if** $r = 2$ **then** $B_i := B_i^2$
**end for**

**Theorem 1.** *Algorithm* TwoBins *is* 2*-competitive for the unweighted online knapsack problem.*

*Proof.* Let $T$ be a set of items, and $OPT(T)$ be the (offline) optimal value for $T$. If $s(T) \leq 1$, then we have $s(B_n^1) = OPT(T) = s(T)$ and $s(B_n^2) = 0$, where $s(A) = \sum_{e \in A} s(e)$ for $A \subseteq T$. Thus the competitive ratio is $OPT(T)/(s(B_n^1) + s(B_n^2))/2) = 2$. Otherwise (i.e., $s(T) > 1$), we have $s(B_n^1) + s(B_n^2) > 1$, which immediately implies that the competitive ratio is $\frac{OPT(T)}{(s(B_n^1)+s(B_n^2))/2} < 2$. Then the algorithm is at most 2-competitive. Moreover, by considering the case in which $s(T) \leq 1$, we can conclude that the algorithm is at least 2-competitive. $\square$

We next show that the ratio in Theorem 1 is tight.

**Theorem 2.** *There exists no randomized online algorithm with competitive ratio less than* 2 *for the unweighted online knapsack problem.*

*Proof.* We use Yao's principle [15]. We construct the following family of input distributions parametrized by a positive integer $n$.

For a given $n$, the probability distribution of the input sequence is as follows:

$$\frac{1}{2} + \varepsilon,\ \frac{1}{2} + \frac{\varepsilon}{2}, \ldots,\ \frac{1}{2} + \frac{\varepsilon}{k},\ \frac{1}{2} - \frac{\varepsilon}{k} \qquad \text{with probability } 1/n \quad (k = 1, \ldots, n), \tag{1}$$

where we identify the items by their sizes (i.e., values), and $\varepsilon$ is a sufficiently small positive number. Then, we note that the optimal expected profit is 1 since the optimal profit of each sequence is $(\frac{1}{2} + \frac{\varepsilon}{k}) + (\frac{1}{2} - \frac{\varepsilon}{k}) = 1$.

For a positive integer $l$, let $A$ denote an deterministic online algorithm that accepts the $l$th item (i.e., the item with size $\frac{1}{2} + \frac{\varepsilon}{l}$) if it is contained in the input sequence. Then Algorithm

$A$ rejects all the items with size $\frac{1}{2} + \frac{\varepsilon}{i}$ for positive integer $i \neq l$. We can see that the expected profit of Algorithm $A$ is at most

$$\frac{1}{2} \cdot \frac{l-1}{n} + 1 \cdot \frac{1}{n} + \left(\frac{1}{2} + \frac{\varepsilon}{l}\right) \cdot \frac{n-l}{n} \leq \left(\frac{1}{2} + \varepsilon\right) \cdot \frac{n+1}{n}.$$

Therefore, the competitive ratio is at least

$$\frac{1}{\left(\frac{1}{2} + \varepsilon\right) \cdot \frac{n+1}{n}},$$

which goes to 2 as $n$ and $\varepsilon$ respectively approach to $\infty$ and 0. $\square$

# 3 Unweighted Removable Online Knapsack Problem

In this section, we consider removable knapsack problem when the value of each item is equal to its size.

## 3.1 Upper bound for randomized competitive ratio

In this subsection, we propose a randomized 10/7-competitive online algorithm for unweighted removable online knapsack problem. Recall that the problem is deterministic $\frac{1+\sqrt{5}}{2}$-competitive, and hence it does not admit deterministic 10/7-competitive algorithm. For example, consider two input sequences $(0.69, 0.4)$ and $(0.69, 0.4, 0.6)$, where we identify items with size (i.e., $(0.69, 0.4)$ denotes that input sequence consists of two items such that the first and the second items respectively have size 0.69 and 0.4). Then in order to obtain deterministic 10/7-competitive algorithm, we must reject 0.4 for the input sequence $(0.69, 0.4)$, since $0.69/0.4 > 10/7$ and moreover, we must reject 0.69 for the input sequence $(0.69, 0.4, 0.6)$, since $(0.6+0.4)/0.69 > 10/7$. They are impossible for any deterministic algorithm. On the other hand, our (randomized) algorithm randomly chooses the one among two deterministic algorithms, where the one rejects 0.4 and the other rejects 0.69.

Our algorithm partitions all the items into three groups, *small*, *medium* and *large* where an item $e$ is called *small*, *medium*, and *large* if $s(e) \leq 0.3$, $0.3 < s(e) < 0.7$, and $s(e) \geq 0.7$, respectively. Let $S$, $M$, and $L$ respectively denote the sets of small, medium, and large items. $M$ is further partitioned into four subsets $M_i$ for $1 \leq i \leq 4$, where $M_1$, $M_2$, $M_3$, and $M_4$ respectively denote the set of the items $e$ with size $0.3 < s(e) \leq 0.4$, $0.4 < s(e) \leq 0.5$, $0.5 < s(e) < 0.6$, and $0.6 \leq s(e) < 0.7$ (see Fig. 1). An item $e$ is also called an $M_i$-*item* if $e \in M_i$.

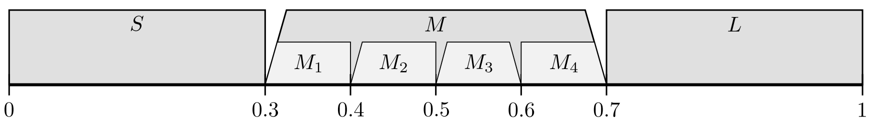


Figure 1: Item partition for our randomized 10/7-competitive online algorithm.

Our algorithm A is briefly described as follows. If a large item comes, A keeps it in the knapsack (by removing all the items we have chosen), since it ensures 10/7-competitivity of the algorithm. Otherwise, we simulate two deterministic subroutines $A_1$ and $A_2$, where we keep the items in the knapsack by following the one of $A_1$ and $A_2$ chosen randomly in advance. Both subroutines first handle medium items (differently) and then choose small items from the largest to the smallest.

Subroutine $A_1$ first chooses the smallest $M_4$-item if it exists. Otherwise, it chooses the smallest $M_3$-item. It then chooses the other items from the largest to the smallest. On the other hand, Subroutine $A_2$ keeps a set of items $I$ with sufficiently large profit, namely, if either (i) $I$ satisfies $0.9 \leq s(I) \leq 1$ or (ii) some $M_4$-item has already come and $I$ satisfies $0.8 \leq s(I) \leq 1$.

Otherwise, $\mathrm{A}_2$ first chooses the smallest $M_2$- and $M_1$-items and then choose the medium items from the smallest to the largest, and the small items from the largest to the smallest in the current knapsack.

Let $e_i$ be the item given in the $i$th round. Define by $B_i$ the set of selected items at the end of the $i$th round by Algorithm A. For $r = 1, 2$, define by $B_i^r$ the set of selected items at the end of the $i$th round by Subroutine $\mathrm{A}_r$. Let $f_i$ denote a flag such that $f_i = 1$ if some $M_4$-item has come by the end of the $i$th round, and $f_i = 0$, otherwise. Then our algorithm A is represented as follows.

---

Algorithm A

---

$B_0,\ B_0^1,\ B_0^2 := \emptyset,\ f_0 := 0$
choose $r$ uniformly at random from $\{1, 2\}$
**for each** item $e_i$ in order of arrival **do**
  **if** $e_i$ in $L$ **then**
    choose it by cancelling all the items in the knapsack, and stop handling the future items.
  **end if**
  **if** $e_i \in M_4$ **then**
    $f_i := 1$
  **else**
    $f_i := f_{i-1}$
  **end if**
  simulate two subroutines $\mathrm{A}_1(f_i, B_{i-1}^1, e_i)$ and $\mathrm{A}_2(f_i, B_{i-1}^2, e_i)$;
  **if** $r = 1$ **then** $B_i := B_i^1$
  **if** $r = 2$ **then** $B_i := B_i^2$
  **if** the expected profit $(s(B_i^1) + s(B_i^2))/2$ is at least 0.7 **then** stop handling the future items.
**end for**

---

Subroutine $\mathrm{A}_1$

---

**if** $f_i = 0$ **then** choose the smallest $M_3$-item from $B_{i-1}^1 \cup \{e_i\}$;
**else** (i.e., $f_i = 1$): choose the smallest $M_4$-item from $B_{i-1}^1 \cup \{e_i\}$.
choose the items among $B_{i-1}^1 \cup \{e_i\}$ from the largest to the smallest.

---

Then we have the following theorem.

**Theorem 3.** *Algorithm A is* $10/7$*-competitive for the unweighted removable online knapsack problem.*

Due to space limitation, we omit the proof of the above theorem.

## 3.2 Lower bound for randomized competitive ratio

Babaioff *et al.* [2] provided a lower bound 5/4 for the randomized competitive ratio of the *general* removable online knapsack problem. In this subsection, we show that 5/4 is also a lower bound even for the unweighted case. The proof is based on Yao's principle. We consider the following input distribution:

$$\begin{cases} 2/3+\varepsilon,\ 1/3,\ 2/3 & \text{(with probability } 1/2\text{)}, \\ 2/3+\varepsilon,\ 1/3, & \text{(with probability } 1/2\text{)} \end{cases} \tag{2}$$

where we identify the items with their size (value) and $\varepsilon$ is a sufficiently small positive number.

**Theorem 4.** *There exists no randomized online algorithm with competitive ratio less than* $5/4$ *for the unweighted removable online knapsack problem.*

Subroutine $\mathrm{A}_2$

**if** $f_i = 0$ and $B^2_{i-1} \cup \{e_i\}$ contains a set of items $I$ with $0.9 \le s(I) \le 1$ **then**
  $B^2_i := I$
**else if** $f_i = 1$ and $B^2_{i-1} \cup \{e_i\}$ contains a set of items $I$ with $0.8 \le s(I) \le 1$ **then**
  $B^2_i := I$
**else**
  choose the smallest $M_2$-item from $B^2_{i-1} \cup \{e_i\}$.
  choose the smallest $M_1$-item from $B^2_{i-1} \cup \{e_i\}$.
  choose the rest of medium items among $B^2_{i-1} \cup \{e_i\}$ from the smallest to the largest.
  choose the small items among $B^2_{i-1} \cup \{e_i\}$ from the largest to the smallest.
**end if**

*Proof.* We consider the input distribution in (2). Then, the optimal expected profit is $1 \cdot \frac{1}{2} + \left(\frac{2}{3} + \varepsilon\right) \cdot \frac{1}{2} = \frac{5}{6} + \frac{\varepsilon}{2}$.

Let $A$ be a deterministic online algorithm. If $A$ rejects the second item, the expected profit is at most $2/3 + \varepsilon$. Otherwise (i.e., $A$ takes the second item after removing the first item), the expected profit is at most $1 \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{2}{3}$.

Therefore, the competitive ratio is at least $(5/6 + \varepsilon/2)/(2/3 + \varepsilon)$ which approaches $5/4$ as $\varepsilon \to 0$. □

# 4 General Removable Online Knapsack Problem

In this section, we consider the general removable online knapsack problem.

## 4.1 Upper bound for randomized competitive ratio

We propose a randomized 2-competitive algorithm for the removable online knapsack problem. Our algorithm can be regarded as randomized and online implementation of the well known 2-approximation algorithm [12] for offline problem, which makes use of algorithms Max and Greedy as follows.

Algorithm Max

$B_0 := \emptyset$
**for each** item $e_i$ in order of arrival **do**
  $B_i := \mathrm{argmax}\{v(e) : e \in B_{i-1} \cup \{e_i\}\}$
**end for**

In the algorithms, let $e_i$ be the item given in the $i$th round, and let $B_i$ be the set of selected items at the end of the $i$th round. We denote by $s(B_i)$ the total size of items in $B_i$.

For a set of items $T = \{e_1, e_2, \ldots, e_n\}$, let $OPT(T)$ denote the optimal (offline) profit, and let $\textsc{Max}(T)$ and $\textsc{Greedy}(T)$ respectively denote the profits obtained by Algorithms Max and Greedy.

**Theorem 5.** *The algorithm that runs* Max *and* Greedy *uniformly at random is at most* 2-*competitive.*

*Proof.* By the definitions of Algorithms Max and Greedy, we have $OPT(T) \le \textsc{Max}(T) + \textsc{Greedy}(T)$, since the optimal profit of the (integral) knapsack problem is at most the one of the fractional knapsack problem, which is again at most $\textsc{Max}(T) + \textsc{Greedy}(T)$. Therefore, the competitive ratio is at most

$$\frac{OPT(T)}{\frac{\textsc{Max}(T)+\textsc{Greedy}(T)}{2}} \le 2.$$

□

Algorithm Greedy

$B_0 := \emptyset$
**for each** item $e_i$ in order of arrival **do**
  Let $B_{i-1} \cup \{e_i\} = \{b_1, \dots, b_k\}$ s.t. $\frac{v(b_1)}{s(b_1)} \geq \frac{v(b_2)}{s(b_2)} \geq \cdots \geq \frac{v(b_k)}{s(b_k)}$
  $B_i := \emptyset$
  **for** $j = 1$ to $k$ **do**
    **if** $s(B_i) + s(b_j) \leq 1$ **then** $B_i := B_i \cup \{b_j\}$
  **end for**
**end for**

### 4.2 Lower bound for randomized competitive ratio

We prove the lower bound $1 + 1/e$ on the competitive ratio of the general removable online knapsack problem by using Yao's principle [15]. We consider the following family of input distributions parametrized by a positive integer $n$. Let $(s, v)$ denote an item whose size and value are $s$ and $v$, respectively. For a given $n$, the probabilistic distribution of the input sequence is

$$(1,1),\ \underbrace{(1/n^2, 1/n), \dots,\ (1/n^2, 1/n)}_{k \text{ items}} \text{ with probability } p_k\ (k = 1, 2, \dots, n^2) \tag{3}$$

where $p_k = \frac{1-e^{-1/n}}{1-e^{-n}} \cdot e^{-(k-1)/n}$.

**Theorem 6.** *There exists no randomized online algorithm with competitive ratio less than* $1+1/e$ *for the removable online knapsack problem.*

*Proof.* We consider the input distribution given in (3). Then we have the optimal expected profit

$$\sum_{k=1}^{n} 1 \cdot p_k + \sum_{k=n+1}^{n^2} \frac{k}{n} \cdot p_k \geq \frac{1-e^{-1/n}}{1-e^{-n}} \cdot n \left( \int_0^1 e^{-t} dt + \int_1^n t e^{-t} dt \right).$$

Let $A$ be a deterministic algorithm for the online knapsack problem. Then it is not difficult to see that $A$ takes the first item $(1,1)$ to have the constant competitive ratio. Let $l$ denote the number of items $(\frac{1}{n^2}, \frac{1}{n})$ that $A$ rejects before item $(1,1)$ is cancelled. Then, the expected profit of the algorithm $A$ is at most

$$\sum_{k=1}^{l} p_k + \sum_{k=l+1}^{n^2} \frac{k-l}{n} \cdot p_k \leq \frac{1-e^{-1/n}}{1-e^{-n}} \cdot n e^{2/n}.$$

Therefore, by using Yao's principle, the competitive ratio is at least

$$\frac{\frac{1-e^{-1/n}}{1-e^{-n}} \cdot n \left( \int_0^1 e^{-t} dt + \int_1^n t \cdot e^{-t} dt \right)}{\frac{1-e^{-1/n}}{1-e^{-n}} \cdot n e^{2/n}} \to 1 + 1/e \quad (n \to \infty)$$

for any randomized online algorithm. □

## Acknowledgment

This work is partially supported by NSFC (11101065) and "the Fundamental Research Funds for the Central Universities" (DUT12LK09) in China, by Grant-in-Aid for Scientific research from the Ministry of Education, Culture, Sports, Science and Technology of Japan, by the Global COE "The Research and Training Center for New Development in Mathematics", and by JST, ERATO, Kawarabayashi Large Graph Project.

## References

[1] M. Babaioff, J. D. Hartline, and R. D. Kleinberg. Selling banner ads: Online algorithms with buyback. In *Proceedings of the 4th Workshop on Ad Auctions*, 2008.

[2] M. Babaioff, J. D. Hartline, and R. D. Kleinberg. Selling ad campaigns: Online algorithms with cancellations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 61–70, 2009.

[3] M. Babaioff, N. Immorlica, D. Kempe, R. Kleinberg, M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 16–28, 2007.

[4] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *Proceedings of the 13th Annual European Symposium*, Lecture Note in Computer Science, pages 689–701. Springer Berlin Heidelberg, 2005.

[5] B. Dean, M. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the sixteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 395–404. Society for Industrial and Applied Mathematics, 2005.

[6] B. Dean, M. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

[7] X. Han, Y. Kawase, and K. Makino. Online knapsack problem with removal cost. In *Computing and Combinatorics*, volume 7434 of *Lecture Notes in Computer Science*, pages 61–73. Springer Berlin Heidelberg, 2012.

[8] X. Han and K. Makino. Online minimization knapsack problem. In *Approximation and Online Algorithms*, volume 5893 of *Lecture Notes in Computer Science*, pages 182–193. Springer Berlin Heidelberg, 2010.

[9] H. Ito, S. Kiyoshima, and Y. Yoshida. Constant-time approximation algorithms for the knapsack problem. In *Theory and Applications of Models of Computation*, pages 131–142, 2012.

[10] K. Iwama and S. Taketomi. Removable online knapsack problems. In *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 293–305. Springer Berlin Heidelberg, 2002.

[11] K. Iwama and G. Zhang. Optimal resource augmentations for online knapsack. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques.*, volume 4627 of *Lecture Notes in Computer Science*, pages 180–188. Springer Berlin Heidelberg, 2007.

[12] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer Berlin Heidelberg, 2004.

[13] G. S. Lueker. Average-case analysis of off-line and on-line knapsack problems. *Journal of Algorithms*, 29(2):277–305, 1998.

[14] A. Marchetti-Spaccamela and C. Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68:73–104, 1995.

[15] A. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227. IEEE, 1977.

[16] Y. Zhou, D. Chakrabarty, and R. Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. *Internet and Network Economics*, pages 566–576, 2008.