Theoretical Computer Science 609 (2016) 185–196

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Online minimization knapsack problem ☆

Xin Han [a,b,*,1], Kazuhisa Makino [c]

[a] Software School, Dalian University of Technology, China
[b] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China
[c] Research Institute for Mathematical Sciences, Kyoto University, 606-8502 Japan

ARTICLE INFO

Article history:
Received 6 November 2014
Received in revised form 17 September 2015
Accepted 19 September 2015
Available online 28 September 2015
Communicated by A. Marchetti-Spaccamela

Keywords:
Knapsack
Competitive analysis
Online algorithms

ABSTRACT

In this paper, we address the online minimization knapsack problem, i.e., the items are given one by one over time and the goal is to minimize the total cost of items that covers a knapsack. We study the *removable* model, where it is allowed to remove old items from the knapsack in order to accept a new item. We obtain the following results.

(i) We propose a 8-competitive deterministic and memoryless algorithm for the problem, which contrasts with the result for the online maximization knapsack problem that no online algorithm has a bounded competitive ratio [14].
(ii) We propose a 2e-competitive randomized algorithm for the problem.
(iii) We derive a lower bound of 2 for deterministic algorithms for the problem.
(iv) We propose a 1.618-competitive deterministic algorithm for the case in which each item has size equal to its cost, and show that this is best possible.

1. Introduction

The knapsack problem is one of the most classical and studied problems in combinatorial optimization and has a lot of applications in the real world [15]. The (classical) knapsack problem is: given a set of items with profits and sizes, and the capacity value of a knapsack, maximize the total profit of selected items in the knapsack satisfying the capacity constraint. This problem is also called the *maximization* knapsack problem (Max-Knapsack). Many variants and generalizations of the knapsack problem have been investigated [15]. Among them, the *minimization* knapsack problem (Min-Knapsack) is one of the most natural ones (see [1,2,5,6] and [15, pp. 412–413]), that is, given a set of items associated with costs and sizes, and the size of a knapsack, minimize the total cost of selected items that cover the knapsack.

In this paper, we focus on the online version of problem Min-Knapsack. Here, the "online" means that items are given over time, i.e., after a decision of rejection or acceptance is made on the current item, the next item is given, and once an item is rejected or removed, it cannot be considered again. The goal of the online minimization knapsack problem is the same as the offline version, i.e., to minimize the total cost.

☆ The previous version of this paper was appeared in conference WAOA 2009 [7]. This work was done when the first author worked in The University of Tokyo and supported in part by the Scientific Grant in Aid of the Ministry of Education, Science, Sports and Culture of Japan.

* Corresponding author.
E-mail addresses: hanxin.mail@gmail.com (X. Han), makino@kurims.kyoto-u.ac.jp (K. Makino).

1 Partially supported by NSFC (11571060), RGC (HKU716412E), the Fundamental Research Funds for the Central Universities (DUT15LK10).

http://dx.doi.org/10.1016/j.tcs.2015.09.021

Table 1
The current status of the complexity of problems Max-Knapsack and Min-Knapsack.

			Max-Knapsack		Min-Knapsack	
			lower bound	upper bound	lower bound	upper bound
offline			FPTAS [12]		FPTAS [1]	
online	non-removable	general	unbounded [13]		**unbounded**	
		size = cost	unbounded [13]		**unbounded**	
	removable	general	unbounded [14]		**2**	**8** **2*e*** (randomized)
		size = cost	1.618 [13]	1.618 [13]	**1.618**	**1.618**

Related work: It is well-known that offline Max-Knapsack and Min-Knapsack both admit a fully polynomial time approximation scheme (FPTAS) [1,6,15]. As for the online maximization knapsack problem, it was first studied using average case analysis by Marchetti-Spaccamela and Vercellis [18]. They proposed a linear-time approximation algorithm such that the expected difference between the optimal and the approximation solution value is $O(\log^{3/2} n)$ under the condition that the capacity of the knapsack grows proportionally to n, the number of items. Lueker [17] further improved the expected difference to $O(\log n)$ under a fairly general condition on the distribution. In 2002, Iwama and Taketomi [13] studied the problem using worst case analysis. They obtained a 1.618-competitive algorithm for the online Max-Knapsack under the *removable* condition, if each item has its size equal to its profit. Here the *removable* condition means that it is allowed to remove some items in the knapsack in order to accept a new item. They also showed that this is best possible by providing a lower bound 1.618 for this case. For the general case, Iwama and Zhang [14] showed that no algorithm for online Max-Knapsack has a bounded competitive ratio, even if the removal condition is allowed. Some generalizations of the online Max-Knapsack such as resource augmentations and Multi Knapsacks were investigated [14,20,11]. Han and Makino studied online max-knapsack problems with limited cuts and got an optimal upper bound [8], and the techniques can be applied to min-knapsack problem with limited cuts. For the randomized removable online knapsack problems, the upper bound is 2 and lower bound is 1.368 [4,10]. There are also some references on online knapsack with buyback [9], i.e., we have to pay some cost when an item in the knapsack is removed for accepting a new item.

Our results: In this paper, we study the online minimization knapsack problem. We first show that no algorithm has a bounded competitive ratio, if the *removable* condition is not allowed. Under the removable condition, we propose two *deterministic* algorithms for the online Min-Knapsack. The first one is simple and has competitive ratio $\Theta(\log \Delta)$, where Δ is the ratio of the maximum size to the minimum size in the items, and the second one has competitive ratio 8. This constant-competitive result for the online Min-Knapsack contrasts with the result for the online Max-Knapsack that no online algorithm has a bounded competitive ratio [14], which is possibly surprising since the problems Max-Knapsack and Min-Knapsack are expected to have the same behavior from a complexity viewpoint (see Table 1).

The first algorithm is motivated by the observation: if all the items have the same size, then a simple greedy algorithm (called *Lowest Cost First* strategy) of picking items with the lowest cost first provides an optimal solution. We partition the input into $\lceil \log \Delta \rceil + 1$ subsets by their *size*. When a new item d_t is given, the algorithm guesses the optimal value within a $O(1)$ approximation factor, by using only the items in the knapsack together with the new item d_t. Then for each subset, we choose items by the Lowest Cost First strategy such that the total cost in each subset is at most $O(1)$ times the optimal value.

In order to improve the above algorithm, it has to select items with the lower total cost. However, this makes it difficult to guess the optimal value, since the item removed cannot be reused, even for guessing the optimal value. We devise the following strategy to overcome this difficulty. At each time, i) we guess the optimal value within $O(1)$ factor, by repeatedly solving fractional Max-Knapsack problems, which is to maximize the total size subject to bounded costs with respect to the items in the knapsack and the coming item, and ii) in order to find items to be kept, for each $j \geq 0$ we construct a subset H_j of items by solving the fractional Max-Knapsack problem subject to 2^{2-j} times the optimal cost; we keep items in $\bigcup_{j\geq 0} H_j$. We guarantee that each class H_j has cost at most 2^{2-j} times the optimal cost, which implies that the total cost in the knapsack is $O(1)$ times the optimal cost. Since a feasible solution of the Min-Knapsack problem is always kept in the knapsack, the procedure above leads to an $O(1)$-competitive algorithm.

We also show that no deterministic online algorithm can achieve competitive ratio less than 2, and provide a *randomized* online algorithm with competitive ratio $2e \approx 5.44$. We finally consider the case in which each item has cost equal to its size. Similarly to the online Max-Knapsack problem [13], we show that the online Min-Knapsack problem admits a 1.618-competitive deterministic algorithm which matches the lower bound.

Table 1 summarizes the current status of the complexity of the problems Max-Knapsack and Min-Knapsack, where the bold letters represent the results obtained in this paper.

The rest of the paper is organized as follows. Section 2 gives definitions of the online Min-Knapsack problem, and shows that the "removable" condition is necessary for the online Min-Knapsack problem. Section 3 presents algorithms for the online Min-Knapsack problem, and Section 4 gives a lower bound 2 for the online Min-Knapsack problem. Finally, in Section 5, we consider the case where each item has cost equal to its size.