

# The Online Knapsack Problem with Departures

Bo Sun<sup>#\*</sup>    Lin Yang<sup>#†</sup>    Mohammad Hajiesmaili<sup>‡</sup>    Adam Wierman<sup>§</sup>  
John C.S. Lui<sup>¶</sup>    Don Towsley<sup>||</sup>    Danny H.K. Tsang<sup>\*\*</sup>

March 16, 2023

## Abstract

The online knapsack problem is a classic online resource allocation problem in networking and operations research. Its basic version studies how to pack online arriving items of different sizes and values into a capacity-limited knapsack. In this paper, we study a general version that includes *item departures*, while also considering *multiple knapsacks* and *multi-dimensional item sizes*. We design a threshold-based online algorithm and prove that the algorithm can achieve order-optimal competitive ratios. Beyond worst-case performance guarantees, we also aim to achieve near-optimal average performance under typical instances. Towards this goal, we propose a data-driven online algorithm that learns within a policy-class that guarantees a worst-case performance bound. In trace-driven experiments, we show that our data-driven algorithm outperforms other benchmark algorithms in an application of online knapsack to job scheduling for cloud computing.

---

<sup>#</sup>These authors contribute equally to this work.

<sup>\*</sup>The Chinese University of Hong Kong. Email: [bsun@cse.cuhk.edu.hk](mailto:bsun@cse.cuhk.edu.hk).

<sup>†</sup>Nanjing University. Email: [linyang@nju.edu.cn](mailto:linyang@nju.edu.cn).

<sup>‡</sup>University of Massachusetts Amherst. Email: [hajiesmaili@cs.umass.edu](mailto:hajiesmaili@cs.umass.edu).

<sup>§</sup>California Institute of Technology. Email: [adamw@caltech.edu](mailto:adamw@caltech.edu).

<sup>¶</sup>The Chinese University of Hong Kong. Email: [cslui@cse.cuhk.edu.hk](mailto:cslui@cse.cuhk.edu.hk).

<sup>||</sup>University of Massachusetts Amherst. Email: [towsley@cs.umass.edu](mailto:towsley@cs.umass.edu).

<sup>\*\*</sup>The Hong Kong University of Science and Technology and The Hong Kong University of Science and Technology (Guangzhou). Email: [eetsang@ust.hk](mailto:eetsang@ust.hk).

# 1 Problem Statement

We study the classic the online knapsack problem (OKP). In its most basic version, there is only one knapsack, and each item is characterized by its value and one-dimensional (scalar) size. The problem is to irrevocably decide whether to admit each item upon its arrival with the goal of maximizing the total values of admitted items while respecting the capacity of the knapsack. The sequence of items can only be revealed one-by-one and may even be adversarial. In this paper, we focus on a novel generalization that includes item departures, while also considering multiple knapsacks and multi-dimensional item sizes.

Consider  $K$  knapsacks in a slotted time horizon  $[T] = \{1, \dots, T\}$ , where each knapsack  $k \in [K]$  has capacity  $C_k \in \mathbb{R}^+$ . A total of  $N$  items arrive sequentially and each item  $n$  is characterized by its item information  $\mathcal{I}_n = \{a_n, \{w_{nk}, v_{nk}, \mathcal{T}_{nk}\}_{k \in [K]}\}$ , where  $a_n$  is the arrival time, and for each knapsack  $k$ ,  $w_{nk}$  and  $v_{nk}$  are the size and value, and  $\mathcal{T}_{nk} := \{s_{nk}, \dots, s_{nk} + d_{nk} - 1\}$  is the set of time slots that item  $n$  requests to stay in knapsack  $k$  from its starting time  $s_{nk}$  to its departure time  $s_{nk} + d_{nk} - 1$ . The set  $\mathcal{T}_{nk}$  contains  $d_{nk}$  consecutive time slots and we call  $d_{nk}$  the duration of the item.

Upon arrival of item  $n$ , a decision maker observes its item information  $\mathcal{I}_n$  and determines (i) whether to admit this item, and (ii) which knapsack this item should be assigned to if it is admitted. Let  $\mathbf{x}_n = \{x_{nk}\}_{k \in [K]}$  denote the decision variable, where  $x_{nk} \in \{0, 1\}$  indicates whether to admit item  $n$  to knapsack  $k$  and  $\sum_{k \in [K]} x_{nk} = 0$  represents declining the item. The goal is then to design an online algorithm to causally determine  $\mathbf{x}_n$  based on the item information up to  $n$ , i.e.,  $\{\mathcal{I}_{n'}\}_{n' \leq n}$ , that maximizes the total value of all admitted items while ensuring the capacities of all knapsacks not to be violated over the time horizon.

Let  $\mathcal{I} := \{\mathcal{I}_n\}_{n \in [N]}$  denote an instance of the online knapsack with departures (OKD). Given  $\mathcal{I}$ , the offline problem is shown as

$$\max_{x_{nk}} \sum_{n \in [N]} \sum_{k \in [K]} v_{nk} x_{nk}, \quad (1a)$$

$$\text{s.t.} \quad \sum_{n \in [N]: t \in \mathcal{T}_{nk}} w_{nk} x_{nk} \leq C_k, \forall k \in [K], t \in [T], \quad (1b)$$

$$\sum_{k \in [K]} x_{nk} \leq 1, \forall n \in [N], \quad (1c)$$

$$x_{nk} \in \{0, 1\}, \forall n \in [N], k \in [K]. \quad (1d)$$

Let  $\text{OPT}(\mathcal{I})$  and  $\text{ALG}(\mathcal{I})$  denote the values obtained by the offline problem (1) and an online algorithm under the instance  $\mathcal{I}$ . The competitive ratio of the online algorithm is defined as the worst-case performance ratio of the offline and online algorithms, i.e.,  $\text{CR} = \max_{\mathcal{I}} \text{OPT}(\mathcal{I}) / \text{ALG}(\mathcal{I})$ . Our goal is to design an online algorithm that can achieve the smallest competitive ratio.

We make following technical assumptions.

**Assumption 1 (Value density fluctuation)** *The value density of each item  $n$  in knapsack  $k$  is bounded, i.e.,  $v_{nk} / (w_{nk} d_{nk}) \in [1, \theta_k], \forall n \in [N]$  and value density ratio is defined as  $\theta_k$ .*

**Assumption 2 (Duration fluctuation)** *The duration of each item  $n$  in knapsack  $k$  is bounded, i.e.,  $d_{nk} \in [\underline{D}_k, \overline{D}_k], \forall n \in [N]$  and duration ratio is defined as  $\alpha_k = \overline{D}_k / \underline{D}_k$ .*

**Assumption 3 (Upper bound of item size)** *The size of each item  $n$  is upper bounded, i.e.,  $w_{nk} \leq \varepsilon_k \leq C_k, \forall k \in [K]$ .*

---

**Algorithm 2** Online Threshold-based Algorithm for Admission Control (OTA)

---

- 1: **input:** item information  $\{v, w, \mathcal{T}\}$ , threshold function  $\phi$ , utilization  $\{z_t\}_{t \in \mathcal{T}}$ , capacity  $C$ ;
  - 2: **output:** admission decision  $\hat{x}$ ;
  - 3: determine a threshold value  $\Phi = \sum_{t \in \mathcal{T}} w\phi(z_t)$ ;
  - 4: **if**  $v \geq \Phi$  and  $z_t + w \leq C, \forall t \in \mathcal{T}$  **then**
  - 5:   item is admissible and set  $\hat{x} = 1$ ;
  - 6: **else**
  - 7:   item is inadmissible and set  $\hat{x} = 0$ .
  - 8: **end if**
- 

## 2 Algorithms & Main Results

---

**Algorithm 1** Online Algorithms for Online Multiple Knapsacks with Departures (OA-OKD)

---

- 1: **input:** threshold function  $\phi = \{\phi_k\}_{k \in [K]}$ , knapsack capacities  $\{C_k\}_{k \in [K]}$ ;
  - 2: **output:** admission and assignment decision  $\mathbf{x}_n = \{x_{nk}\}_{k \in [K]}$ ;
  - 3: **initialization:** initial knapsack utilization  $z_{kt}^{(0)} = 0, \forall k, t$ ;
  - 4: **for** each item  $n$  with item information  $\mathcal{I}_n$  **do**
  - 5:   **for** each knapsack  $k \in [K]$  **do**
  - 6:     call Algorithm 2 to check admissibility  $\hat{x}_{nk} = \text{OTA}(\mathcal{I}_n, \phi_k, \{z_{kt}^{(n-1)}\}_{t \in \mathcal{T}_{nk}}, C_k)$ ;
  - 7:   **end for**
  - 8:   **if**  $\sum_{k \in [K]} \hat{x}_{nk} > 0$  **then**
  - 9:     admit item  $n$  and assign it to knapsack  $k' = \arg \max_{k \in [K]: \hat{x}_{nk}=1} v_{nk}$ ;
  - 10:     set  $x_{nk'} = 1$  and  $x_{nk} = 0, \forall k \in [K] \setminus \{k'\}$ ;
  - 11:   **else**
  - 12:     decline item  $n$  and set  $x_{nk} = 0, \forall k \in [K]$ ;
  - 13:   **end if**
  - 14:   update knapsack utilization  $z_{kt}^{(n)} = z_{kt}^{(n-1)} + w_{nk}x_{nk}, \forall k \in [K], t \in \mathcal{T}_{nk}$ .
  - 15: **end for**
- 

We propose a simple yet effective online algorithm (OA-OKD) to solve OKD in Algorithm 1. It consists of two parts: decomposing the multiple knapsack problem into the admissibility check of each individual knapsack and admission control of each individual knapsack via an online threshold-based algorithm (OTA). The key step is the admission control of items to each knapsack by calling the OTA subroutine in Algorithm 2. To check admissibility, OTA defines a threshold value (line 3) as  $\Phi = \sum_{t \in \mathcal{T}} w\phi(z_t)$ , where  $\phi(z_t)$  can be interpreted as the marginal cost of the unit item if it stays in the knapsack in slot  $t$ , and is a function of the real-time knapsack utilization  $z_t$ . Since  $\text{OTA}_\phi$  is fully parameterized by  $\phi$ , the key design question is how to determine the threshold function  $\phi$  such that OA-OKD is competitive with the offline optimum.

By carefully designing the threshold function, our main results can be summarized as follows.

**Theorem 1** *Under Assumptions 1-3, there exists a parameter  $\gamma_k = O(\ln(\alpha_k \theta_k)), \forall k \in [K]$ , if the item size is upper bounded by  $\varepsilon_k \leq C_k \ln 2 / \gamma_k, \forall k \in [K]$ , and the threshold function is given by  $\phi^\gamma := \{\phi_k^{\gamma_k}\}_{k \in [K]}$ , where*

$$\phi_k^{\gamma_k}(z) = \exp(z\gamma_k/C_k) - 1, z \in [0, C_k], \forall k \in [K], \quad (2)$$

*then the competitive ratio of the algorithm OA( $\phi^\gamma$ ) is  $O(\ln(\alpha\theta))$ , where  $\theta = \max_{k \in [K]} \theta_k$  and  $\alpha = \max_{k \in [K]} \alpha_k$ .*

**Theorem 2** *There is no online algorithm that can achieve a competitive ratio smaller than  $\Omega(\ln(\alpha\theta))$  for the online multiple one-dimensional knapsacks with departures.*

Combining the upper bound result in Theorem 1 and the lower bound result in Theorem 2, we conclude that our proposed  $\mathbf{OA}(\phi^\gamma)$  achieves an order-optimal competitive ratio for  $\mathbf{OKD}$ . In the full paper [1], this proposed algorithm can be further extended to consider the multi-dimensional item size and also achieve the order-optimal competitive ratio. In addition, we additionally design a data-driven online algorithm that can adaptively select the parameter  $\gamma$  such that the overall algorithm can work well on practical instances and, in the meanwhile, provide the worst-case guarantees.

## References

- [1] B. Sun, L. Yang, M. Hajiesmaili, A. Wierman, J. C. Lui, D. Towsley, and D. H. Tsang. The online knapsack problem with departures. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–32, 2022.