Competitive Algorithms for Online Multidimensional Knapsack Problems

LIN YANG, University of Massachusetts Amherst, USA
ALI ZEYNALI, University of Massachusetts Amherst, USA
MOHAMMAD H. HAJIESMAILI, University of Massachusetts Amherst, USA
RAMESH K. SITARAMAN, University of Massachusetts Amherst & Akamai Technologies, USA
DON TOWSLEY, University of Massachusetts Amherst, USA

In this paper, we study the online multidimensional knapsack problem (called OMdKP) in which there is a knapsack whose capacity is represented in m dimensions, each dimension could have a different capacity. Then, n items with different scalar profit values and m-dimensional weights arrive in an online manner and the goal is to admit or decline items upon their arrival such that the total profit obtained by admitted items is maximized and the capacity of knapsack across all dimensions is respected. This is a natural generalization of the classic single-dimension knapsack problem and finds several relevant applications such as in virtual machine allocation, job scheduling, and all-or-nothing flow maximization over a graph. We develop two algorithms for OMdKP that use linear and exponential reservation functions to make online admission decisions. Our competitive analysis shows that the linear and exponential algorithms achieve the competitive ratios of $O(\sqrt{\theta\alpha})$ and $O(\log(\theta\alpha))$, respectively, where α is the ratio between the aggregate knapsack capacity and the minimum capacity over a single dimension and θ is the ratio between the maximum and minimum item unit values. We also characterize a lower bound for the competitive ratio of any online algorithm solving OMdKP and show that the competitive ratio of our algorithm with exponential reservation function matches the lower bound up to a constant factor.

CCS Concepts: • **Theory of computation → Online algorithms**.

ACM Reference Format:
Lin Yang, Ali Zeynali, Mohammad H. Hajiesmaili, Ramesh K. Sitaraman, and Don Towsley. 2021. Competitive Algorithms for Online Multidimensional Knapsack Problems. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 3, Article 30 (December 2021), 30 pages. https://doi.org/10.1145/3491042

1 INTRODUCTION

The online knapsack problem [15] (OKP) is a classical online optimization problem that has application in a variety of domains such as cloud and edge computing [35, 53], online admission control [11], online routing of virtual switches [38], and control of distributed energy resources in smart grids [1–3, 42]. In the basic version of OKP, an online algorithm must make irrevocable decisions about which items with different values and weights to pack into a capacity-limited knapsack without knowing what items will arrive in the future. The goal of the algorithm is to maximize the aggregate value of admitted items while respecting the capacity of the knapsack.

Authors' addresses: Lin Yang, University of Massachusetts Amherst, USA; Ali Zeynali, University of Massachusetts Amherst, USA; Mohammad H. Hajiesmaili, University of Massachusetts Amherst, USA; Ramesh K. Sitaraman, University of Massachusetts Amherst & Akamai Technologies, USA; Don Towsley, University of Massachusetts Amherst, USA.

2476-1249/2021/12-ART30 $15.00
https://doi.org/10.1145/3491042

This problem has been tackled using the competitive online algorithm framework [8], and there are algorithms [13, 15, 54] that achieve a competitive ratio of $O(\log\theta)$ for the basic version, where θ is a value fluctuation ratio between the most and the least valuable items. Further, it has been shown that the competitive ratio has a lower bound of $\Omega(\log\theta)$ [54], hence the algorithms in [52, 54] are optimal since their competitive ratio is tight.

Recently, the basic OKP has been extended to better capture properties of real-world applications. In [13], an extended version of OKP, an online multiple knapsack problem has been considered where there are multiple knapsacks with different capacities, and divisible items could be packed into a subset of knapsacks. Hence, the question becomes whether to admit or reject, and if admitted, how to pack the item into a subset of knapsacks. In other work [44], the problem has been extended to mechanism design settings, and game-theoretic properties such as truthfulness have been investigated. Motivated by a cloud resource pricing application, the authors in [52] extended OKP to a time-expanded version where the knapsack slots are expanded over time and items can demand for a subset of time slots . In [51], the basic setting is extended to knapsacks with packing costs. We will review the related work in more detail in Section 7.

In this paper, we study a different extension of OKP, the online multidimensional knapsack problem (OMdKP), in which there is a single knapsack whose capacity is represented by an m-dimensional vector, and the weights (or sizes) of online items are m-dimensional. The goal of an online decision maker is to pack the most valuable items so that the capacity of the knapsack in each dimension is not exceeded. Note that this problem differs from the online multiple knapsack problem in which fixed-weight items are packed into one or multiple knapsacks. In other words, in the online multiple knapsack problem decisions must be made about both *admission* and *allocation* whereas in OMdKP, only an admission decision is required. In Section 7, we further explore the connections and differences between OMdKP and several other related problems such as online multidimensional bin packing [10].

The multidimensional knapsack problem is of significant practical relevance since it applies to several application scenarios. As an example, consider a scenario in which there are different resources (e.g., CPU, memory, storage), and each arriving item (e.g., jobs or virtual machines) can request more than one resource. Hence, OMdKP has been extensively studied in the literature [22, 23, 39, 47, 49]. However, all these works tackle the problem in an offline setting. In this work, we focus on an online version where the goal is to design competitive algorithms for OMdKP.

The high-level idea of our algorithm design is to devise an online reservation (a.k.a. threshold) function for admitting online items, a technique that is also commonly used in OKP. The exact characterization of reservation function such that it leads to an online algorithm with bounded competitive ratio, however, is a challenging task and highly depends on the problem setting and underlying assumptions, i.e., infinitesimal weights, or known value fluctuation ratio. Specifically, we devise a reservation function that associates an *implicit admission cost* to the knapsack as a function of its utilization. The reservation function is an increasing function of utilization, i.e., the higher the knapsack's utilization, the higher the admission cost. Then, given a proper reservation function, an online strategy simply admits an incoming item only if its value is not less than the current admission cost calculated from the reservation function. This high-level idea has been used to develop optimal reservation functions for the basic OKP [52, 54] and the multiple knapsack version [43], and it has been shown that the corresponding online algorithms can achieve the best competitive ratios. However, designing online reservation policies for OMdKP is much more challenging than for OKP, and no existing algorithms for OKP can be applicable in our setting. Technically speaking, in OMdKP, the additional challenge for designing a reservation function originates from multiple dependent dimensions of the knapsack that should be properly captured by a reservation function. More specifically in OMdKP, while each item is associated with a single scalar value, its weight is multidimensional and the weight vector could be unbalanced across different dimensions. A naïve idea of extending existing algorithms for OKP by using the aggregate weight across all dimensions will not be adequate, since items with the same value and aggregate weight will be treated the same. For instance, consider two items with the same values and same aggregate weight, one larger in an over-utilized scarce dimension, while the other item is larger in an under-utilized dimension. A good online algorithm for OMdKP should admit the latter item more readily than the former.

Our Contributions. In this paper, we develop two classes of online algorithms for OMdKP by using two different reservation policies, and analyze their competitive ratios. Specifically, we design two reservation policies explicitly accounting for item weights across different dimensions. The first reservation policy is a linear function and the second is an exponential function of knapsack utilization and the corresponding online algorithms exhibit different competitive ratios. The main contribution of this paper is providing the first order-optimal algorithm for OMdKP under the commonly-made "infinitesimal" assumption that item weights are "small enough"' as compared to the capacity of the knapsack.

We develop two algorithms based on linear (LinRP) and exponential (ExpRP) reservation policies, and characterize their competitive ratios. The competitive ratios are characterized as a function of two parameters α and θ. We define $\alpha_j = C/C_j$ and $\alpha = \max_j \alpha_j$, where C_j is the capacity of dimension j and C is the aggregate capacity over all dimensions. In other words, α is the ratio between the aggregate knapsack capacity and the minimum single dimension capacity. The second parameter is θ that refers to the ratio between the maximum and minimum unit values of all items, where unit value is the ratio between the value of an item and its aggregate weight over all dimensions. Our analysis shows that LinRP achieves a competitive ratio of $O(\sqrt{\theta\alpha})$, and the competitive ratio of ExpRP is $O(\log \theta\alpha)$. Then, we derive a lower bound of $\Omega(\log \theta\alpha)$ for the competitive ratio of any online algorithm solving OMdKP. Hence, ExpRP attains an order-optimal competitive ratio. Lastly, we extend the ExpRP algorithm to the fractional version of OMdKP and our analysis shows the fractional exponential algorithms achieves the competitive ratio of $\max\{8, 4\log\theta\alpha\} + 1$. It is also worth noting that different from majority of prior work, our theoretical analysis of the competitive ratio of the proposed algorithms provides explicit bounds on the ratio of the maximum size of items to the capacity of the knapsack. This appeared as the bounds of values of ϵ in Theorems 3 and 4.

It is worth mentioning that when the values of θ and α are sufficiently large, the competitive ratio of ExpRP is better than that of LinRP. However, in practical scenarios with a small number of dimensions, the linear algorithms may outperform the exponential ones; hence both algorithms are practically relevant. In Section 6, we numerically evaluate the performance of both LinRP and ExpRP, and compare their performance with some extensions of single-dimension knapsack algorithms.

2 THE ONLINE MULTIDIMENSIONAL KNAPSACK PROBLEM

In this section, we present the online multidimensional knapsack problem (OMdKP) as a generalization of the classic online knapsack problem. In OMdKP, there is a knapsack with m ($m \geq 2$) dimensions, and items arrive in an online fashion with different weights (or, sizes) along each dimension, and the goal is to pack as many as possible high-valued items such that the capacity constraint of the knapsack over different dimensions is respected. This is a natural generalization of the knapsack problem and is motivated by several real-world applications such as online job-resource allocation [7, 26], all-or-nothing flow maximization [17], and more. In the following, we formally introduce OMdKP.

Table 1. Summary of notations

Notation	Description
n	Number of items, indexed by i
m	Number of dimensions, indexed by j
C_j	Capacity of dimension j
C	Aggregate capacity over all dimensions
α	Ratio of the aggregate capacity to the minimum capacity, i.e., $\alpha = C/\min_j C_j$
α_j	Ratio of the aggregate capacity to the capacity of dimension j, i.e., $\alpha = C/C_j$
v_i	Value of item i
$\mathbf{w_i} = [w_{i,1}, \ldots, w_{i,j}, \ldots, w_{i,m}]$	Weight vector of the i-th item
w_i	Aggregate weight of the i-th item
p_i	Unit value of the i-th item
θ	Variation in value, i.e., ratio between the maximum and minimum item values
$\boldsymbol{u}_i = [u_{i,1}, \ldots, u_{i,j}, \ldots, u_{i,m}]$	Knapsack utilization after the decision on the i-th item
x_i	The optimization variable on the admission decision of item i
ω	An input instance to OMdKP
Ω	The set of all feasible input instances to OMdKP
A	An online algorithm for OMdKP
A(ω)	Profit of algorithm A under instance ω
OPT(ω)	Profit of the optimal algorithm under instance ω
CR(A)	The competitive ratio of algorithm A defined in Equation (4)

2.1 Problem Statement

We consider a knapsack whose capacities along m dimensions is represented by vector $\mathbf{C} = [C_1, \ldots, C_j, \ldots, C_m]$, where C_j represents the capacity of dimension $j \in [m] = \{1, \ldots, m\}$ and C is the aggregate capacity over all dimensions, i.e., $C = \sum_{j\in[m]} C_j$. Without loss of generality, we assume, $C_1 \leq C_2 \leq \cdots \leq C_m$. Items arrive in an online fashion, each with a different value and weights. Specifically, in round $i \in [n] = \{1, \ldots, n\}$, item i arrives with value $v_i \geq 0$, and a weight vector $\mathbf{w_i} = [w_{i,1}, \ldots, w_{i,j}, \ldots, w_{i,m}]$, where $w_{i,j} \geq 0$ is the size of item i in dimension j of the knapsack. Given item values and weights along with the capacity vector of the knapsack, the offline version of OMdKP can be formulated as

$$[\texttt{OMdKP}] \quad \max \sum\nolimits_{i\in[n]} v_i x_i, \quad \text{s.t.,} \sum\nolimits_{i\in[n]} w_{i,j} x_i \leq C_j, \forall j \in [m], \tag{1}$$

where x_i's are the optimization variables and $x_i = 1$ if item i is admitted and $x_i = 0$, otherwise. We consider both integral and fractional versions of the problem. In the fractional version, $x_i \in [0,1], \forall i \in [n]$, and $x_i \in \{0,1\}, \forall i \in [n]$ for the integral version. We are interested in an online setting in which items arrive one-by-one and an online algorithm has to immediately decide whether to admit the incoming item without knowing the future and in the absence of a stochastic modeling. We present our main results for the integral version of OMdKP. However, our results can be extended to the fractional case as we present in details in Section 5.

2.2 Additional Notations and Assumptions

To facilitate our algorithm design, we introduce an auxiliary variable to represent the knapsack utilization in each dimension after an online algorithm makes an admission decision for item i. In particular, let $\boldsymbol{u}_i = [u_{i,1}, \ldots, u_{i,j}, \ldots, u_{i,m}]$ be the knapsack utilization after making a decision to admit item i or not, where $0 \leq u_{i,j} \leq C_j$ corresponds to the utilization of dimension j up to the i-th round, i.e., the aggregate size of admitted items up to item i for the integral version. For convenience, $u_{0,j} = 0$. We define p_i as the *unit value* of item i, i.e.,

$$p_i := \frac{v_i}{w_i}, \forall i \in [n], \tag{2}$$

where $w_i = \sum_{j\in[m]} w_{i,j}$ is the aggregate size of item i. We further assume that $p_i \in [p_{\min}, p_{\max}], \forall i \in [n]$, where $p_{\min}$ and $p_{\max}$ are lower and upper bound values on the unit value of each item. We define $\theta = p_{\max}/p_{\min}$ as the value fluctuation ratio. To capture the variations in capacity, we define parameter α as the ratio between the aggregate capacity of knapsack over all dimensions and the minimum single-dimension capacity, i.e., $\alpha = \sum_j C_j / \min_j C_j$. For the ease of analysis, we also define $\alpha_j = C/C_j$. Both parameters θ and α play a critical role in the competitive analysis of the proposed algorithms.

In our algorithms, we assume that normalized weights of items are much smaller than the capacity, i.e., $w_{i,j}/C_j \leq \varepsilon \ll 1, \forall i, j$, where ε is defined as the largest single-dimension normalized weight of items, that is

$$\varepsilon := \max_{i\in[n]} \max_{j\in[m]} \frac{w_{i,j}}{C_j}. \tag{3}$$

This assumption naturally holds in large-scale systems and is common in online knapsack literature [43, 50]. We present our results by explicit characterization of valid ranges for ε. Also, in Section 5, we relax this assumption for our algorithms for the fractional model.

2.3 Competitive Algorithm Design Framework

Our goal is to design an online algorithm that makes an irrevocable admission decision based on the available information, i.e., the knapsack capacity and the current utilization. The goal of an online algorithm is to perform nearly as well as the offline optimum. We conduct our analysis using the competitive framework [8] with *competitive ratio* as the performance metric. Specifically, for an online algorithm $\mathtt{A}$, the competitive ratio is

$$\mathtt{CR}(\mathtt{A}) = \max_{\omega\in\Omega} \frac{\mathtt{OPT}(\omega)}{\mathtt{A}(\omega)}, \tag{4}$$

where $\omega \in \Omega$ denotes a feasible instance to $\mathtt{OMdKP}$ and Ω is the set of all feasible instances to $\mathtt{OMdKP}$. Also, $\mathtt{OPT}(\omega)$ is the offline optimum under instance ω, and $\mathtt{A}(\omega)$ is the profit obtained by executing online algorithm $\mathtt{A}$ over instance ω. Note that the capacity of knapsack along m dimensions for the online algorithm is identical to the knapsack capacity for the offline algorithm. We present our algorithms for $\mathtt{OMdKP}$ in Section 3, followed by the competitive analysis in Section 4, and an algorithm and analysis for the fractional setting in Section 5.

3 ONLINE ALGORITHMS

In this section, we present two online algorithms for $\mathtt{OMdKP}$, and characterize their competitive ratio as a function of α, i.e., the capacity variation parameter, and θ, the value variation parameter. We note that assuming known θ is required for designing an online algorithm with bounded competitive ratio even for the basic version of online knapsack problem [13, 15, 43, 50, 54]. The parameter α, however, is new in the multidimensional setting and captures the heterogeneity of capacity of dimensions.

In the following, we first motivate algorithm design for OMdKP by analyzing the competitiveness of the First-come First-serve (FCFS) strategy and show FCFS is $O(\theta\alpha)$-competitive. Hence, we need to design better algorithms with better competitive ratios.

Then, we present the LinRP and ExpRP reservation policies as two online algorithms that take into account the current knapsack utilization in admitting items. We analyze the performance of both algorithms in Section 4 and show that they achieve $O(\sqrt{\theta\alpha})$ and $O(\log\theta\alpha)$ competitive ratios, respectively. We also characterize a logarithmic lower bound for the competitive ratio of online algorithms solving OMdKP, and show that the competitive ratio of ExpRP matches this lower bound asymptotically. The tight competitive analysis of ExpRP is the major theoretical contribution of this paper.

3.1 Warm-up: First-Come First-Serve: An $O(\theta\alpha)$-Competitive Algorithm

As a baseline algorithm, we consider the First-Come-First-Serve algorithm (FCFS), which admits each arriving item unless there is insufficient space. The following theorem with a proof in Appendix A.1 shows that FCFS is at least $\Omega(\theta\alpha)$-competitive.

Theorem 1. *The competitive ratio of* FCFS *is* $\Omega(\theta\alpha)$.

The above result shows that FCFS which is oblivious to item values and the residual capacity of individual dimensions, fails to achieve a good competitive ratio. To design online algorithms with improved competitive ratios, our idea is to balance the residual capacity of different dimensions by assigning implicit cost functions to each dimension as a function of their residual capacity. Based on this high-level intuition, in the following, we introduce two policies that respectively associate linear and exponential reservation functions with the dimensions by which the algorithm is able to evaluate costs of admitting incoming items based on their demand and the available space. With the above construction, the proposed algorithms admit an incoming item only when its value is larger than or equal to the current admission cost.

3.2 Online Algorithm 1: A Linear Reservation Policy (LinRP)

We first introduce an $O(\sqrt{\theta\alpha})$-competitive algorithm, called as the Linear Reservation Policy (LinRP). Recall that the high-level idea of our algorithm design is to determine the implicit admission cost, and admit the incoming item if its value is higher than or equal to the admission cost. The admission cost is an increasing function of current utilization, the higher the utilization, the higher the admission cost. Based on the current utilization of knapsack, LinRP defines $z_{i,j}, j \in [m]$ as the normalized utilization of dimension j after the arrival of the i-th item as follows:

$$z_{i,j} = \left\lfloor \frac{u_{i,j}}{C_j}\sqrt{\theta m} \right\rfloor, j \in [m]. \tag{5}$$

Upon arrival of item i, LinRP admits the item only if its value satisfies the following threshold.

$$v_i \geq \max_{j\in[m]} z_{i-1,j}\sqrt{\frac{2\alpha_j}{m}}w_{i,j}, \tag{6}$$

where $z_{i-1,j}$ is the normalized utilization of dimension j *before* arrival of item i. Details of the LinRP algorithm are summarized in Algorithm 1. As compared to the naive FCFS strategy, dynamically adjusts a threshold with respect to $z_{i,j}$ to admit an item based on the scarcity across each dimension. In doing so, the LinRP algorithm can reserve the scarce space for high-valued items in future.

We now proceed to provide insights behind the reservation policy in Equation (6). The left hand side is the value of the incoming item i, and $z_{i-1,j}\sqrt{\alpha_j/m)}$ in the right hand side represents scarcity of the space in dimension j. Scarcity increases linearly with $z_{i-1,j}$, hence the name linear reservation

Algorithm 1 The `LinRP` Algorithm, upon arrival of item i

1: **if** $v_i \geq \max_{j\in[m]} z_{i-1,j}\sqrt{\frac{2\alpha_j}{m}}w_{i,j}$ and $w_{i,j} \leq C_j - u_{i-1,j}, j \in [m]$ **then** ▹ `admit item` i `if item value is greater than the admission cost and there is enough space.`
2: $x_i \leftarrow 1$ ▹ `admit item`
3: **else**
4: $x_i \leftarrow 0$ ▹ `decline item`
5: **end if**
6: $u_{i,j} \leftarrow u_{i-1,j} + x_i w_{i,j}, j \in [m]$ ▹ `update utilization`
7: $z_{i,j} \leftarrow \left\lfloor \frac{u_{i,j}}{C_j}\sqrt{\theta m}\right\rfloor, j \in [m].$ ▹ `update normalized utilization`

policy. By multiplying the scarcity factor and the item weight in the same dimension, the `LinRP` algorithm evaluates the cost in the corresponding dimension to admit item i. The item is admitted if and only if there is enough space and its value is larger than or equal to all evaluated dimension costs. Intuitively, in order to be admitted, items demanding scarce dimensions should have larger values. In this way, `LinRP` prevents saturation of scarce dimensions by low-valued items. This leads to an improved competitive ratio of `LinRP` as compared to the `FCFS` policy.

3.3 Online Algorithm 2: An Exponential Reservation Policy (`ExpRP`)

Now, we proceed to introduce `ExpRP` that uses the same high-level idea of `LinRP`, but, with different definitions for the normalized utilization and the reservation function for evaluating the admission cost. The new definition of the normalized cost $z_{i,j}$ is

$$z_{i,j} = \left\lfloor \frac{u_{i,j}}{C_j} \log\left(\theta\alpha_j\right)\right\rfloor, j \in [m], \tag{7}$$

that represents the normalized utilization of each dimension after arrival of the i-th item. Then, the new item i is admitted if there is enough capacity and the following inequality holds.

$$v_i \geq \sum_{j=1}^{m} \left(2^{z_{i-1,j}} - 1\right) w_{i,j}. \tag{8}$$

The details of the `ExpRP` algorithm are summarized in Algorithm 2. We note that one may attain a different competitive ratio by specifying a different constant instead of 2 in the reservation function (8). For simplicity, we just choose 2 as the base of the reservation function. It is worth noting that following the recent approach in designing data-driven online algorithms [50], this coefficient could be changed to a parameter that could be learned in real-time to improve practical performance.

Given enough available space for admission, `ExpRP` makes an admission decision based on Equation (8), in which factor $(2^{z_{i-1,j}} - 1)$ represents the scarcity of dimension j. The larger the variable $z_{i-1,j}$, the larger the scarcity factor. By multiplying the scarcity and the weight $w_{i,j}$ in the same dimension, `ExpRP` evaluates the cost in each dimension, and admits item i only if its value is larger than or equal to the aggregate cost over all dimensions. Compared to `LinRP`, the scarcity factor in `ExpRP` increase exponentially in $z_{i,j}$ and `ExpRP` admits the item only when its value is at least equal to the sum of the costs over all dimensions. In general, `ExpRP` algorithm is more conservative than `LinRP` in its admission decisions and thus tends to reserve the remaining capacity for higher-valued items. In the next section, we analyze the competitive ratios of both `LinRP` and `ExpRP`.

Algorithm 2 The ExpRP Algorithm, upon arrival of item i

1: **if** $v_i \geq \sum_{j=1}^{m} (2^{z_{i-1,j}} - 1)\, w_{i,j}$ and $w_{i,j} \leq C_j - u_{i-1,j}, j \in [m]$ **then** ▹ admit item i if item value is greater than the admission cost and there is enough space.
2: $\quad x_i \leftarrow 1$ ▹ admit item
3: **else**
4: $\quad x_i \leftarrow 0$ ▹ decline item
5: **end if**
6: $u_{i,j} \leftarrow u_{i-1,j} + x_i w_{i,j}, j \in [m]$ ▹ update utilization
7: $z_{i,j} \leftarrow \left\lfloor \frac{u_{i,j}}{C_j} \log\left(\theta \alpha_j\right) \right\rfloor, j \in [m],$ ▹ update normalized utilization

4 COMPETITIVE ANALYSIS

We first present our lower and upper bound results, and provide several remarks and insights about our results. The proofs are given in Section 4.2.

4.1 Main Results

We present a lower bound for any competitive algorithm providing a feasible solution for OMdKP (in Theorem 2) followed by the competitive results for LinRP (in Theorem 3) and ExpRP (in Theorem 4). Recall that ε defined in Equation (3) serves as an upper bound of ratios between single-dimension size of items and the capacity of the knapsack. This valid range for this parameter is explicitly characterized to guarantee the competitive ratios.

Theorem 2. *(Lower Bound on Competitive Ratio for* OMdKP*) The competitive ratio of any online algorithm providing a feasible solution to* OMdKP *is* $\Omega(\log \theta\alpha)$.

Theorem 3. *With* $\varepsilon < 1/(2\sqrt{m})$ *and* $m \geq 4$*, the competitive ratio of* LinRP *satisfies*

$$\mathtt{CR(LinRP)} \leq \sqrt{2\alpha}\left(\left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon\right)\frac{\left\lfloor\sqrt{\theta m}\right\rfloor\left(\left\lfloor\sqrt{\theta m}\right\rfloor - 1\right)}{\theta\sqrt{m}}\right)^{-1}.$$

Remark. When $\varepsilon \longrightarrow 0$, representing the case with arbitrarily small item sizes, we have

$$\mathtt{CR(LinRP)} \leq \sqrt{2\alpha}\frac{\theta^{3/2} m}{\left\lfloor\sqrt{\theta m}\right\rfloor\left(\left\lfloor\sqrt{\theta m}\right\rfloor - 1\right)}, \text{ or } \mathtt{CR(LinRP)} = O\left(\sqrt{\theta\alpha}\right).$$

Theorem 4. *With* $\varepsilon < \min\{1/3, 1/(2\log(\theta\alpha))\}$*, the competitive ratio of* ExpRP *satisfies*

$$\mathtt{CR(ExpRP)} \leq \max\left\{12, \frac{4\log(\theta\alpha)}{1 - 2\varepsilon\log(\theta\alpha)}\right\} + 1.$$

Remarks. (1) When $\varepsilon \longrightarrow 0$, the competitive ratio of ExpRP satisfies

$$\mathtt{CR(ExpRP)} \leq \max\{12, 4\log\theta\alpha\} + 1.$$

Hence, ExpRP is $O(\log(\theta\alpha))$-competitive.

(2) Comparing the result in theorems 2 and 4 shows that ExpRP achieves the optimal competitive ratio up to a constant factor.

(3) For unit capacities for all dimensions, and unit values, i.e., $\theta = 1$, we have the modified $\varepsilon < \min\{1/3, 1/(2\log m)\}$, and the competitive ratio of ExpRP satisfying

$$\mathtt{CR(ExpRP)} \leq \max\left\{12, 4\frac{\log m}{1 - 2\varepsilon\log m}\right\} + 1.$$

(4) When comparing the competitive ratios of reLR and ExpRP, one finds that ExpRP outperforms LinRP when α and/or θ are large. However, when these values are small, LinRP may outperform ExpRP. In practice, however, these values are small, and this is further investigated in our numerical experiments in Section 6.

(5) In the special case with $m = 1$, OMdKP is reduced to the basic version of online knapsack problem [15, 54], or equivalently the so-called one-way trading [21] (see [14] for the equivalence). Correspondingly, the ExpRP algorithm is reduced to the optimal algorithm for those two problems which uses exponential thresholds to admit items.

(6) As we mentioned in the system model, we consider the aggregate size of an item to be simply the sum of its sizes over all dimensions, i.e., $w_i = \sum_{j\in[m]} w_{i,j}, \forall i \in [n]$. Our results can be extended to account for a weighted aggregate size over dimensions. Specifically, let $d_j \geq 0,\ \forall j \in [m]$, be a priority coefficient associated with dimension j and hence we can redefine the size of item i as $w_i = \sum_{j\in[m]} d_j w_{i,j}$. In this new setting, we can extend the results by redefining $C = \sum_{j\in[m]} d_j C_j$, and $\alpha_j = C/(d_j C_j)$. Then, similar competitive ratios can be obtained by setting $\alpha = \sum_{j\in[m]} d_j C_j / \min_{j\in[m]} d_j C_j$.

(7) In the literature, primal-dual based methods have been used to design and analyze online algorithms in many other related settings [12, 43]. Either primal-dual based method or ours has the potential to achieve the best result. Actually, many of those algorithms, such as [43], make decisions also based on a "pseudo-price function", which is similar to the threshold functions in our both LinRP and ExpRP algorithms.

4.2 Proofs

In this section, we prove the results in theorems 2-4.

An Informal Proof for Theorem 2. We first provide an informal, yet, intuitive proof sketch for Theorem 2. A complete proof is provided in Appendix A.3. Note that the OMdKP problem can be seen as an extension of its unit-density version of OMdKP (the OMdKP problem with $\theta = 1$) or the one-way trading problem [21] (with $m = 1$), then the adversary can construct cases where the competitive ratio of any online algorithm is either $\Omega(\log\alpha)$ (lower bound for the unit-density version of OMdKP, see **step 1** in the detailed proof of Theorem 2 in Appendix A.3) or $\Omega(\log\theta)$ (lower bound for the one-way trading problem). By combining the two adversaries in the above two problems, we can easily prove a lower bound for the general OMdKP problem which is $\Omega(\log\theta\alpha)$, since

$$\mathsf{CR}(\mathsf{A}) \geq \max\{\Omega(\log\alpha), \Omega(\log\theta)\} \geq \frac{1}{2}\Omega(\log\alpha) + \frac{1}{2}\Omega(\log\theta) = \Omega(\log\theta\alpha).$$

A proof of Theorem 3. Let $\mathbf{z} = [z_1, z_2, \ldots, z_m]$ be the final state of the system executing the LinRP algorithm, where $z_j = z_{n,j}, j \in [m]$. Let $\mathcal{J}_l$, $l = \{0, 1, 2, \ldots, \lfloor\sqrt{\theta m}\rfloor\}$, be the set of dimensions satisfying $z_j \geq l$.

We prove the result by analyzing the two cases, (1) $C_j - u_{n,j} \geq \varepsilon C_j, \forall j \in [m]$, representing the case that by the end of running the algorithm, the knapsack is not saturated along any dimension; and (2) $C_j - u_{n,j} < \varepsilon C_j$, for some $j \in [m]$, representing that at least one dimension is almost saturated.

Case 1: $C_j - u_{n,j} \geq \varepsilon C_j, \forall j \in [m]$. In this case, we can guarantee that the remaining space is always larger than or equal to εC_j and thus all of items will be admitted if Equation (6) is satisfied. Consider the following constraints for incoming item i

$$\text{constraint} - j : v_i \geq z_j \sqrt{\frac{2\alpha_j}{m}} w_{i,j},\ j \in \mathcal{J}_1. \tag{9}$$

We categorize items violating the j-th constraint as Type-j items and define n_j as the number of admitted Type-j items and i_k, $k = \{1, 2, \ldots, n_j\}$, be the k-th admitted Type-j item by any algorithm. By the definition and using Equation (9), we upper bound the aggregate value of admitted Type-j items, denoted as V_j, as follows

$$V_j = \sum_{k=1}^{n_j} v_{i_k} < \sum_{k=1}^{n_j} z_j \sqrt{\frac{2\alpha_j}{m}} w_{i_k,j} = z_j \sqrt{\frac{2C}{mC_j}} \sum_{k=1}^{n_j} w_{i_k,j} \le z_j \sqrt{\frac{2CC_j}{m}},$$

where the first inequality holds by the definition of Type-j items and the second one uses the fact that $\sum_{k=1}^{n_j} w_{i_k,j} \le C_j$.

Considering that $z_{i,j}$ is non-decreasing over time and less than or equal to z_j, if there is an item satisfying all the constraints in (9), it will be admitted by LinRP. Thus, the aggregate value of the admitted items satisfying all the constraints in Equation (9), is not greater than $V_{\texttt{LinRP}}$, as the aggregate value obtained by LinRP. An item admitted by any algorithm is either a Type-j item, $j \in \mathcal{J}_1$, or the one satisfying all the above-mentioned constraints. Then, we can upper bound the cumulative values of admitted items by any algorithm by $\sum_{j\in\mathcal{J}_1} V_j + V_{\texttt{LinRP}}$. We also have $V_{\texttt{LinRP}} \ge \sum_{j\in\mathcal{J}_1} z_j C_j \sqrt{1/(\theta m)}$. Putting together the above results yields

$$\texttt{CR(LinRP)} \le \frac{\sum_{j\in\mathcal{J}_1} V_j + V_{\texttt{LinRP}}}{V_{\texttt{LinRP}}} \le \frac{\sum_{j\in\mathcal{J}_1} z_j \sqrt{\frac{2CC_j}{m}}}{\sum_{j\in\mathcal{J}_1} z_j C_j \sqrt{\frac{1}{\theta m}}} + 1 \le \max_{j\in\mathcal{J}_1} \frac{z_j \sqrt{\frac{2CC_j}{m}}}{z_j C_j \sqrt{\frac{1}{\theta m}}} + 1 \le \sqrt{2\theta\alpha} + 1. \tag{10}$$

Case 2: $C_j - u_{n,j} < \varepsilon C_j$, for some $j \in [m]$.

Combined with the assumption that $\varepsilon < 1/(2\sqrt{\theta m})$, it follows that there is some dimension j' such that the final state $u_{n,j'}$ satisfies $u_{n,j'} > C_{j'} - C_{j'}/(2\sqrt{\theta m})$. Correspondingly, by the discretization step in LinRP, we have $z_{j'} \ge \left\lfloor \sqrt{\theta m} \right\rfloor - 1$. The aggregate value of items admitted by LinRP is

$$\begin{aligned}
V_{\texttt{LinRP}} = \sum_{i=1}^{n} x_i v_i &\ge \sum_{i=1}^{n} x_i \max_j z_{i-1,j} \sqrt{\frac{2\alpha_j}{m}} w_{i,j} \\
&\ge \sum_{i=1}^{n} x_i z_{i-1,j'} \sqrt{\frac{2\alpha_{j'}}{m}} w_{i,j'} \\
&= \sum_{i=1}^{n} z_{i-1,j'} \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2\alpha_{j'}}{m}} \\
&= \sum_{i=1}^{n} \left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \sqrt{\theta m} \right\rfloor \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2\alpha_{j'}}{m}} \\
&\ge \sum_{l=1}^{\left\lfloor \sqrt{\theta m} \right\rfloor - 1} l \left(\frac{C_{j'}}{\sqrt{\theta m}} - 2\varepsilon C_{j'} \right) \sqrt{\frac{2\alpha_{j'}}{m}}.
\end{aligned} \tag{11}$$

The first inequality in the above equation is simply based on the rules of the algorithm. The last inequality uses the feature of the step function and the fact that $u_{i,j'} - u_{i-1,j'} \le \varepsilon C_{j'}$. One can find a proof for it in A.2. Then, we can further lower bound the above equation as follows

$$V_{\texttt{LinRP}} \ge \left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon \right) \frac{\left\lfloor \sqrt{\theta m} \right\rfloor \left(\left\lfloor \sqrt{\theta m} \right\rfloor - 1 \right)}{2\sqrt{m}} \sqrt{2CC_{j'}}.$$

We have

$$
\begin{aligned}
\mathtt{CR(LinRP)} \leq& \theta C \left(\left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon \right) \frac{\left\lfloor \sqrt{\theta m} \right\rfloor \left(\left\lfloor \sqrt{\theta m} \right\rfloor - 1 \right)}{2\sqrt{m}} \sqrt{2CC_{j'}} \right)^{-1} \\
\leq& \sqrt{2\alpha} \left(\left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon \right) \frac{\left\lfloor \sqrt{\theta m} \right\rfloor \left(\left\lfloor \sqrt{\theta m} \right\rfloor - 1 \right)}{\theta \sqrt{m}} \right)^{-1}.
\end{aligned}
\tag{12}
$$

Comparing the results in Equations (10) and (12) completes the proof.

A proof of Theorem 4. We define $\mathbf{Z} = [z_1, z_2, \dots, z_m]$ as the ending state of the system executing the proposed ExpRP algorithm. Let $\mathcal{J}_l$, $l = 0, 1, 2, \dots, \lfloor \log(\theta\alpha) \rfloor$, be the set of dimensions satisfying $z_j \geq l$. Similarly, the proof is executed case by case.

Case 1: $C_j - u_{n,j} \geq \varepsilon C_j$, for any $j \in [m]$.

Considering that the weight of items in dimension j is always less than or equal to εC_j, we have that a job is always admitted when Equation (8) holds.

First, we provide a lower bound for cumulative values of admitted items by the ExpRP algorithm, which is denoted by V_{ExpRP}. Based on the rules of the algorithm, one finds that, for each admitted item, the following equation holds.

$$
v_i \geq \sum_{j=1}^{m} \left(2^{z_{i-1,j}} - 1\right) w_{i,j}.
$$

By the above equation, we can lower bound V_{ExpRP} as follows.

$$
\begin{aligned}
V_{\mathsf{ExpRP}} =& \sum_{i=1}^{n} x_i v_i = \sum_{i=1}^{n} x_i p_i \sum_{j \in \mathcal{J}} w_{i,j} \\
\geq& \sum_{i=1}^{n} \sum_{j \in [m]} \left(2^{z_{i-1,j}} - 1\right) x_i w_{i,j} \\
\geq& \sum_{i=1}^{n} \sum_{j \in [m]} \left(2^{\left\lfloor \frac{u_{i-1,j}}{C_j} \log(\theta\alpha_j) \right\rfloor} - 1 \right) (u_{i,j} - u_{i-1,j}) \\
\geq& \sum_{j \in \mathcal{J}_1} \sum_{l=1}^{z_j - 1} \left(C_j \log^{-1}(\theta\alpha_j) - \varepsilon C_j \right) \left(2^l - 1 \right) \\
=& \sum_{j \in \mathcal{J}_1} \left(C_j \log^{-1}(\theta\alpha_j) - \varepsilon C_j \right) \left(2\left(2^{z_j - 1} - 1 \right) - z_j + 1 \right) \\
=& \sum_{j \in \mathcal{J}_1} \left(C_j \log^{-1}(\theta\alpha_j) - \varepsilon C_j \right) \left(2^{z_j} - z_j - 1 \right),
\end{aligned}
\tag{13}
$$

where the first two inequalities are by the rules of the algorithm, and the last one follows similar lines with the proof in A.2.

In addition to the above, we have another lower bound for V_{ExpRP}

$$
V_{\mathsf{ExpRP}} \geq \sum_{j \in \mathcal{J}_1} C_j \log^{-1}(\theta\alpha_j).
$$

Combining the above two equations yields

$$\begin{aligned} V_{\mathsf{ExpRP}} \geq \max &\left\{ \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right), \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1}\left(\theta\alpha_j\right) - \varepsilon C_j\right)\left(2^{z_j} - z_j - 1\right) \right\} \\ \geq &\beta \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right) + (1-\beta) \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1}\left(\theta\alpha_j\right) - \varepsilon C_j\right)\left(2^{z_j} - z_j - 1\right), \end{aligned} \tag{14}$$

for any $\beta \in [0, 1]$.

Next, we prove an upper bound for the total values of admitted items by any algorithm.

An item i with weight vector $\boldsymbol{w}_i$ is called the Type-I item if the following equation holds.

$$v_i < \sum_{j\in[m]} (2^{z_j} - 1)\, w_{i,j}.$$

Otherwise, an item is called the Type-II item. Obviously, an item is either a Type-I item or a Type-II item. In the following, we will upper bound the aggregate values of Type-I and Type-II items admitted by any algorithm, respectively.

Aggregate value of Type-I items. Let i_k, $k = 1, 2, \ldots, n_1$, be the indices of Type-I items and x_{i_k}, $k = 1, 2, \ldots, n_1$, be the decisions correspondingly. Then, by the definition of the Type-I item, we have that for any i_k, $k = 1, 2, \ldots, n_1$, there is

$$x_{i_k} v_{i_k} < x_{i_k} \sum_{j=1}^{m} (2^{z_j} - 1)\, w_{i_k,j}.$$

It follows from the above equation that

$$\begin{aligned} \sum_{k=1}^{n_1} x_{i_k} v_{i_k} &< \sum_{k=1}^{n_1} x_{i_k} \sum_{j=1}^{m} (2^{z_j} - 1)\, w_{i_k,j} \\ &= \sum_{k=1}^{n_1} \sum_{j=1}^{m} (2^{z_j} - 1)\, x_{i_k} w_{i_k,j} \\ &= \sum_{j=1}^{m} (2^{z_j} - 1) \sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \\ &\leq \sum_{j=1}^{m} (2^{z_j} - 1)\, C_j, \end{aligned}$$

where the last inequality uses the fact that $\sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \leq C_j$, $j = 1, 2, \ldots, m$. By the above equation, we upper bound the aggregate value of Type-I items that are admitted by any algorithm, which is at most $\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j$.

Aggregate value of Type-II items. All the Type-II items are accepted by the ExpRP algorithm since each of them satisfies

$$v_i \geq \sum_{j\in[m]} (2^{z_j} - 1)\, w_{i,j} \geq \sum_{j\in[m]} (2^{z_{i-1,j}} - 1)\, w_{i,j}.$$

Thus, the aggregate value of Type-II items admitted by any algorithm is not larger than that of admitted items by the ExpRP algorithm, i.e., V_{ExpRP}.

Concluding the above results, we upper bound the aggregate value of items admitted by any algorithm by

$$\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j + V_{\mathsf{ExpRP}}.$$

Combined with the results in Equation (14) with $\beta = 1/4$, the competitive ratio of ExpRP satisfies

$$\begin{aligned}
\mathsf{CR}(\mathsf{ExpRP}) &\leq \frac{\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j + V_{\mathsf{on}}}{V_{\mathsf{on}}} \\
&= \frac{\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j}{V_{\mathsf{on}}} + 1 \\
&\leq \frac{\sum_{j\in\mathcal{J}_1} C_j + \sum_{j\in\mathcal{J}_2} (2^{z_j} - 1)\, C_j}{\frac{1}{4}\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right) + \frac{3}{4}\sum_{j\in\mathcal{J}_1}\left(C_j \log^{-1}\left(\theta\alpha_j\right) - \varepsilon C_j\right)\left(2^{z_j} - z_j - 1\right)} + 1 \\
&\leq \max\left\{4\frac{\sum_{j\in\mathcal{J}_1} C_j}{\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right)}, \frac{4\sum_{j\in\mathcal{J}_2} (2^{z_j} - 1)\, C_j}{3\sum_{j\in[m]}\left(C_j \log^{-1}\left(\theta\alpha_j\right) - \varepsilon C_j\right)\left(2^{z_j} - z_j - 1\right)}\right\} + 1 \\
&\leq \max\left\{\max_{j\in\mathcal{J}_1} 4\log\left(\theta\alpha_j\right), \max_{j\in\mathcal{J}_2} 4\frac{\log\left(\theta\alpha_j\right)}{1 - \varepsilon\log\left(\theta\alpha_j\right)}\right\} + 1 \\
&\leq \max\left\{4\log\left(\theta\alpha\right), 4\frac{\log\left(\theta\alpha\right)}{1 - \varepsilon\log\left(\theta\alpha\right)}\right\} + 1,
\end{aligned}$$

where the forth inequality uses the fact that

$$\frac{2^{z_j} - 1}{2^{z_j} - z_j - 1} \leq 3, \text{ for any } j \in \mathcal{J}_2.$$

Case 2: $C_j - u_{n,j} < \varepsilon C_j$, for some $j \in [m]$.

Without loss of generality, we assume that dimension j' satisfies $C_{j'} - u_{n,j'} \geq \varepsilon C_{j'}$. By definition, there is $z_{j'} \geq \lfloor \log\left(\theta\alpha_{j'}\right) \rfloor - 1$.

In the following, we assume $\theta\alpha_{j'} \geq 8$. Otherwise, by using the fact that $\varepsilon \leq 1/3$, we have

$$\mathsf{CR}(\mathsf{ExpRP}) \leq \frac{\theta C}{u_{n,j'}} \leq \frac{\theta C}{(1-\varepsilon)C_{j'}} \leq \frac{\theta C}{\frac{2}{3}C_{j'}} < \frac{3}{2} \times 8 = 12,$$

completing the proof.

By the rules of the ExpRP algorithm, we have

$$
\begin{aligned}
V_{\mathsf{ExpRP}} &= \sum_{i=1}^{n} x_i v_i \\
&\geq \sum_{i=1}^{n} x_i \sum_{j=1}^{m} \left(2^{z_{i-1,j}} - 1\right) w_{i,j} \\
&\geq \sum_{i=1}^{n} \left(2^{z_{i-1,j'}} - 1\right) x_i w_{i,j'} \\
&= \sum_{i=1}^{n} \left(2^{\left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \log^{-1}(\theta\alpha_{j'}) \right\rfloor} - 1\right) \left(u_{i,j'} - u_{i-1,j'}\right) \\
&\geq \sum_{l=1}^{\lfloor \log(\theta\alpha_{j'}) \rfloor - 1} \left(2^l - 1\right) \left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right) \\
&= \left(2\left(2^{\lfloor \log(\theta\alpha_{j'}) \rfloor - 1} - 1\right) - \left\lfloor \log\left(\theta\alpha_{j'}\right) \right\rfloor + 1\right) \left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right) \\
&= \left(2^{\lfloor \log(\theta\alpha_{j'}) \rfloor} - \left\lfloor \log\left(\theta\alpha_{j'}\right) \right\rfloor - 1\right) \left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right) \\
&= \left(\theta\alpha_{j'} - \log\left(\theta\alpha_{j'}\right) - 1\right) \left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right) \geq \frac{\theta\alpha_{j'}}{4} \left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right),
\end{aligned}
$$

where the first inequality uses the rules of the algorithm, and the last one uses the fact that

$$
2^{\lfloor \log(\theta\alpha_{j'}) \rfloor} - \left\lfloor \log\left(\theta\alpha_{j'}\right) \right\rfloor - 1 \geq \frac{1}{2} \cdot 2^{\lfloor \log(\theta\alpha_{j'}) \rfloor} \geq \frac{\theta\alpha_{j'}}{4},
$$

when $\left\lfloor \log\left(\theta\alpha_{j'}\right) \right\rfloor \geq 3$.

Thus,

$$
\mathsf{CR}(\mathsf{ExpRP}) \leq \frac{\theta C}{\frac{\theta\alpha_{j'}}{4}\left(C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}\right)} = \frac{4C_{j'}}{C_{j'} \log^{-1}\left(\theta\alpha_{j'}\right) - 2\varepsilon C_{j'}} \leq \frac{4\log(\theta\alpha)}{1 - 2\varepsilon \log(\theta\alpha)}.
$$

Concluding the above two cases yields

$$
\mathsf{CR}(\mathsf{ExpRP}) \leq \max\left\{12, \frac{4\log(\theta\alpha)}{1 - 2\varepsilon\log(\theta\alpha)}\right\} + 1.
$$

This completes the proof.

5 EXTENSIONS TO FRACTIONAL MODEL WITH ARBITRARY ITEM WEIGHTS

In this section, we extend our algorithms and results to fractional OMdKP, where each item could be admitted partially, i.e., x_i is a real value in $[0, 1]$, and the obtained value is also proportional to the admitted fraction, i.e., $x_i v_i$. We also relax the small size assumption for the fractional case. Recall that the previous algorithms are analyzed by bounding the value of ϵ defined in Equation (3) that captures the limits on item size. For the fractional algorithm design, we relax those assumptions.

We consider the fractional model with arbitrary weights, where each item can be partially packed to a multidimensional knapsack. For brevity, we omit extending the linear reservation policy and only investigate the modified exponential reservation policy for the fractional model, called as ExpRP-F. ExpRP-F determines the admission amount of an item in an iterative manner. Specifically, it splits each incoming item into multiple fractions indicated by parameter y in ExpRP-F and check the exponential admission criterion as used in ExpRP to admit those fractions one-by-one. The

Algorithm 3 The ExpRP-F algorithm for fractional packing of items with arbitrary sizes

Initialization: $x_i \leftarrow 0$, $u_j \leftarrow u_{i-1,j}$, $j \in [m]$
while $x_i \leq 1$ **do**
  $z_j \leftarrow \left\lfloor \frac{u_j}{C_j} \log\left(\theta\alpha_j\right) \right\rfloor$
  $y \leftarrow \min_{j\in[m]} \frac{(z_j+1)C_j \log^{-1}(\theta\alpha_j) - u_j}{w_{i,j}}$
  if $yv_i \geq \sum_{j=1}^{m} (2^{z_j} - 1)\, yw_{i,j}$ **then**
    $x_i \leftarrow \min\{1, x_i + y\}$
    $u_j \leftarrow u_j + yw_{i,j}$, $j \in [m]$
  else
    return x_i
  end if
end while
return x_i

iterative process stops when the item is fully admitted or the admissions criterion that increases the admission cost iteratively violates. ExpRP-F is summarized in Algorithm 3.

Theorem 5. *The competitive ratio of* ExpRP-F *satisfies* $\mathrm{CR}(\texttt{ExpRP-F}) \leq \max\{8, 4\log\theta\alpha\} + 1$.

Comparing the result in Theorem 4 show that ExpRP-F achieves a better competitive ratio than ExpRP. In addition, since ExpRP-F partitions the incoming item into smaller pieces and applies the exponential admission criterion to each piece in an iterative manner, we can relax those bounded item size assumptions for the integral model in the analysis of ExpRP-F. Our proof for Theorem 5 is given in Appendix B.

6 NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments to verify the theoretical results and evaluate the performance of the proposed algorithms. We compare the performance of both linear and exponential policies for OMdKP and several baseline algorithms such as FCFS and some heuristics that are extended version of algorithms for single-dimension knapsack problem (more details in Section 6.2).

6.1 Experimental Setup

As the performance metric, we report the *empirical profit ratio* of different algorithms defined as the ratio between the offline optimal profit and the profit obtained by the online algorithm. Note that the profit ratio is the empirical counterpart of the theoretical competitive ratio. In all experiments, we report the average profit ratios of different algorithms for 20 random trials for 2000 items. Also, we report confidence intervals (shaded areas) as well as cumulative distribution functions (CDF) of all evaluated instances, so more statistical values including the worst-case profit ratios are observable. Unless otherwise mentioned, we set the unit-value fluctuation ratio to be $\theta = 5$, i.e., unit values of items are randomly drawn from $[1, 5]$, $C = m$, $\alpha/m = 2$, and capacities are chosen randomly such that $C_1 \leq C_2 \leq ... \leq C_m$ holds.

6.2 Baseline algorithms

We are not aware of existing solution algorithms for online multidimensional knapsack problems; hence we only compare the proposed algorithms with the offline optimal solution and the following

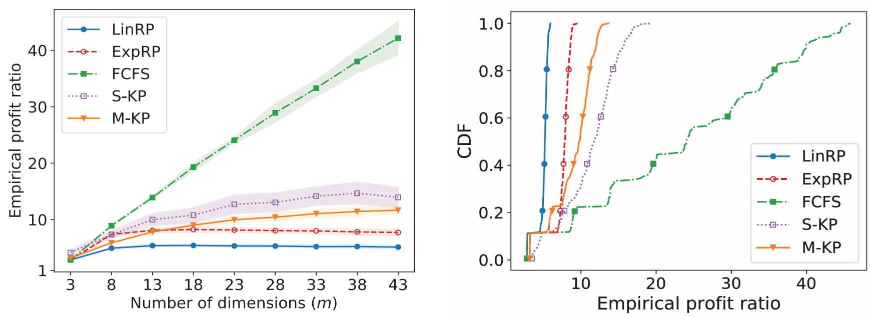

Fig. 1. Comparison of FCFS with LinRP, ExpRP, S-KP, and M-KP with varying number of dimensions m. As m grows, the performance FCFS degrades substantially, but, for the knapsack-based algorithms (both LinRP and ExpRP, and the other two alternatives) the performance degradation is marginal. Due to the poor performance of FCFS, we just compare the performance of LinRP, ExpRP, S-KP, and M-KP in the rest of this section.

baselines. The first baseline is FCFS, which is the simple first-come first-serve policy that does not account for the item values and current utilization of the knapsack in admission decisions.

The other two baselines are heuristics that are simple extensions of existing online algorithms for the single-dimensional knapsack problem. The second algorithm is S-KP, in which the multidimensional inputs, i.e., item weights and capacities, are aggregated into a single scalar value, we then apply the algorithm proposed in [52] and admit an item if the single dimension S-KP admits the item. The third baseline algorithm is M-KP, which runs m independent single-dimension online knapsack algorithm, one for each dimension and admits the item only if all m algorithms admit the item. Indeed both S-KP and M-KP fail to capture unbalanced demands across dimensions and in the following we compare their performance in a range of experimental scenarios.

6.3 Comparison with FCFS

We first compare the performance of LinRP and ExpRP to FCFS, S-KP, and M-KP as a function of number of dimensions. We consider the following scenario for item arrivals. 2000 items arrive in two batches of 1000 each. Items in the first batch have single-dimension demands and items in the second batch have $m/2$-dimension demands. Items in the second batch arrive after the arrival of all of the items in the first batch. For each item in the first or second batch, the set of demanding dimensions has been selected uniformly at random from m dimensions. We set $\theta = 1$ and $\theta = 5$ for the first and second batch of items, respectively. In Figure 1, we vary the number of dimensions and report the average empirical profit ratio of the different algorithms. The results show a substantial increase in the empirical profit ratio of FCFS as m increases, while the profit ratios of both LinRP and ExpRP (and the other two knapsack-based alternatives) increase slightly. Since the performance of FCFS is substantially worse than that of the other algorithms, in the rest of this section, we remove FCFS from future comparisons and focus on comparing the performance of the knapsack-based baseline algorithms, i.e., S-KP and M-KP, to our algorithms.

Another interesting observation is that LinRP outperforms ExpRP in Figure 1, while the theoretical competitive ratio of ExpRP is better than LinRP. We believe this is due to the following two reasons. First, to be worst-case optimal, ExpRP is very conservative in admitting items, and waits for high value and/or low weight items, and this admission threshold increases exponentially. This is aligned with the common understanding of online algorithms that are designed for the worst-case and may not perform well with typical input instances. Second, the difference in the theoretical competitive

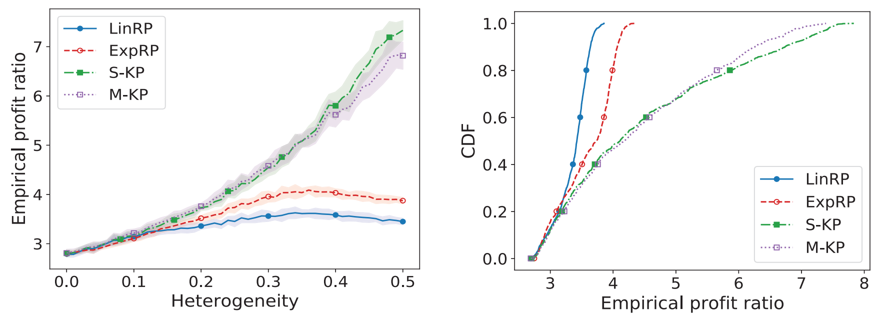

Fig. 2. The impact of heterogeneity (on the size of jobs across different dimensions) on different algorithms. For the definition of heterogeneity we refer to Section 6.4. Heterogeneity equal to 0 reduces our problem to a single knapsack problem, hence, the performance of all algorithms is close to each other. Our algorithms clearly outperform S-KP and M-KP when heterogeneity is high.

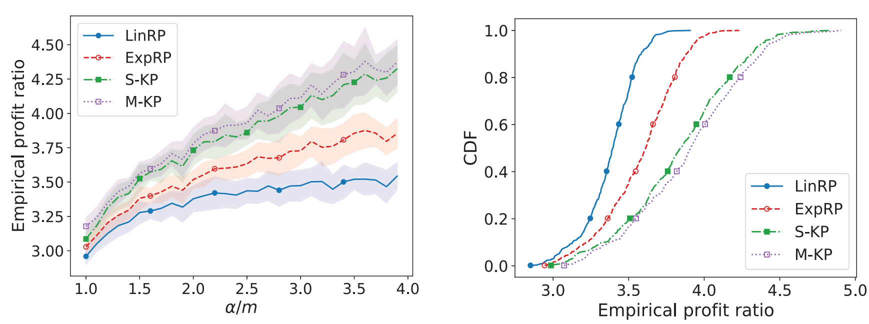

Fig. 3. The impact of α (captures heterogeneity on the capacity of dimensions) on different algorithms. Parameter $\alpha = \sum_j C_j / \min_j C_j$ is the ratio between the aggregate capacity of knapsack and the minimum single-dimension capacity, e.g., $\alpha/m = 1$ represent the case that dimensions have the same capacity.

ratio of LinRP and ExpRP appears when we deal with sufficiently large values of θ and α such that the difference between $\sqrt{.}$ and $\log(.)$ functions becomes clear. In most of our experiments, however, the values of θ and α are small, e.g., $\theta = 5$, $\alpha = 5$. Also, typical values for these parameters in practice are expected to be small. A promising future direction is to integrate both algorithms with the predictive models and data-driven adaptation tools. We discuss these approaches in Section 8.

6.4 The Impact of Heterogeneity among Items

The main motivation for new algorithm design in the multidimensional knapsack originates from the fact that the size of each incoming item across different dimensions might be unbalanced such that the prior algorithms for the single-dimension setting fail to achieve satisfactory performance. In what follows, we design an experiment to investigate the impact of heterogeneity of sizes of items across different dimensions. To create a notion of item heterogeneity, we consider the following experimental scenario. Consider an $m = 20$-dimensional knapsack that is demanded by $n = 2000$ items. Items arrive in two batches of size n_1 and $n_2 = n - n_1$. The first batch consists of one-dimension items and The second batch consists of items demanding at most $m/2$ dimensions

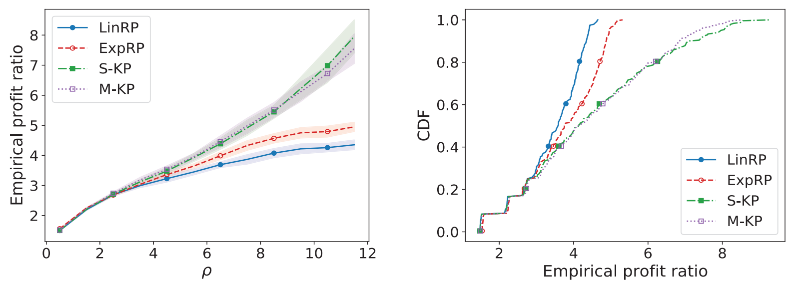

Fig. 4. The impact of different knapsack capacities on different algorithms. Different capacities are represented by parameter $\rho = \sum_{i \in [n]} \sum_{j \in [m]} w_{i,j}/C$, e.g., $\rho = 10$ represents the case that the aggregate item size is 10 times than that of the aggregate capacity of knapsack, i.e., demand is roughly 10 times the available resource.

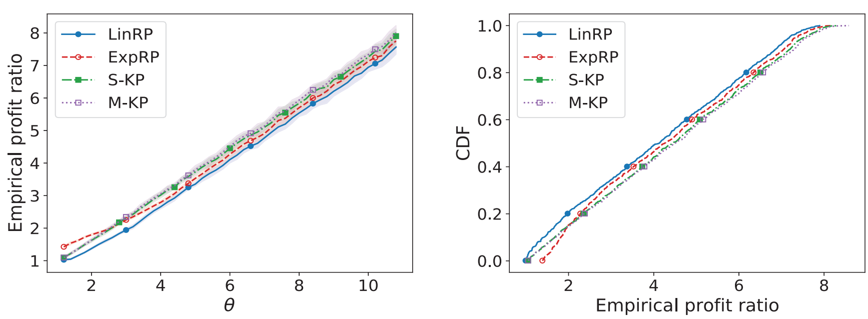

Fig. 5. The impact of parameter θ (captures variability of item values) on different algorithms. Parameter $\theta = p_{\max}/p_{\min}$, and captures the ratio between the maximum and minimum unit-value of items, e.g., $\theta = 10$ represents the case that the most valuable item is 10 times better than the least valuable item.

of the knapsack. The exact number of demanding dimensions for each item has been chosen from a uniform distribution in the range $[3, m/2]$. Heterogeneity is defined as the ratio between n_1 and n_2. In Figure 2, we vary this heterogeneity ratio from 0, i.e., $n_1 = 2000, n_2 = 0$, to 0.5, i.e., $n_1 = 1000, n_2 = 1000$, and report the average empirical profit ratios of the algorithms (on the left) and the CDF (on the right) of `LinRP`, and `ExpRP`, as compared to `S-KP`, `M-KP`. The graphs show that the empirical profit ratios of `S-KP` and `M-KP` rise to 7.9 and 7.4 with heterogeneity of 0.5; however, the empirical ratio of `LinRP` and `ExpRP` never exceed 3.7 and 4.2. The results also demonstrate that the empirical profit ratio of single-dimensional algorithms increases continuously as the heterogeneity increases and hence they fail to effectively admit items. On the other hand, while the profit ratio of `LinRP` and `ExpRP` increases initially with the increase of heterogeneity, it becomes robust after the heterogeneity of ≥ 0.3.

6.5 The Impact of Model Parameters: Capacity, Value Variation θ, and Dimension Capacity Variation α

The theoretical results of our proposed algorithms for `OMdKP` are obtained as functions of different parameters of the problem such as the value fluctuation and dimension capacity variation. These parameters may impact the empirical performance of the proposed algorithms. Hence, as the last experiment, we evaluate the impact of problem parameters, α (Figure 3), knapsack capacity (Figure 4), and θ (Figure 5), on the performance of `LinRP`, `ExpRP`, and the baseline algorithms. To evaluate the impact of capacity, we introduce parameter ρ as the ratio between the aggregate weights of all items and the capacity, i.e., $\rho = \sum_{i\in[n]} \sum_{j\in[m]} w_{i,j}/C$. We use the two batches of items described in Section 6.4 with heterogeneity = 0.2, and set $\theta = 5$, $\alpha = 2$, and $\rho = 5$ as default values. We set $C = m$, and we select capacities of knapsack in each dimension in a way described in Section 6.1. We vary only one parameter in each experiment, maintaining the rest set to default values. Finally, we report the average empirical profit ratios and CDFs of profit ratios of all algorithms. The results are reported in Figures 3, 4, and 5. The results in Figures 3 and 4 demonstrate that when α (normalized to α/m) and ρ increase, the empirical profit ratios of `LinRP` and `ExpRP` increase slightly while the empirical profit ratio for `S-KP` and `M-KP` increases significantly. For example, when ρ reaches 12, the profit ratios of `S-KP` and `M-KP` are close to 8; however, the profit ratios for `LinRP` and `ExpRP` are less than 5. The increase of empirical profit ratios in both cases (the increase of α and ρ) makes sense and it follows from our theoretical results since the competitive ratios of both `LinRP` and `ExpRP` increase with parameter α. We also note that this observation makes sense intuitively since with the increase of both α and ρ, the design space for the offline optimization becomes larger and hence the online algorithms cannot compete effectively with the offline optimum. Last, Figure 5 shows that the profit ratios of all algorithms increase as the item value variation increases, which is aligned with the fact that theoretical competitive ratios increase with θ.

7 RELATED WORK

The offline version of `OMdKP` is a well-studied problem in different settings in literature [22, 23, 29, 39, 47, 49]. Our problem is similar to 0-1 version [9, 23] each admitted item must be packed into the knapsack entirely. The problem has been applied to a some application domains as well, e.g., hardware-software partitioning [27], resource allocation [26]. Nevertheless, to the best of our knowledge, the problem has not studies in the online setting. The offline version of the problem has been studied in [24], where k-dimensional geometric knapsack with a capacity that is represented as a k-dimensional hyperrectangle. Then, some k-dimensional hyperrectangle items arrive sequentially. The goal is to find a set of items that can be placed inside the knapsack without any need to rotate, such that there is no overlap between them and the aggregate value of all admitted items is maximized.

In addition to the prior work for the online version of single-dimension knapsack, this problem has been revisited extensively in literature by adding some additional assumptions to the basic model. For example, in [20], several variants of the basic online knapsack problem have been studied. They explore the sum-objective and max-objective function in which the profit of the knapsack is equal to the value of the maximum valued item placed in the knapsack. However, they restrict the offline algorithm to a unit capacity knapsack while the online algorithm may use a larger capacity. A greedy algorithm (like admit the item if its size does not exceed the size of the most oversized item in the knapsack) is developed for an enhanced item admission. In addition, there are some restrictive modeling assumptions such that the items are categorized into four categories (small, medium, big, huge), and the algorithmic decision is based on the category of the arrived item. Hence, these differences makes our algorithms in the paper in clear contrast with those in [20].

Another category of similar problems is the online multiple knapsack problem (`OMKP`) [13, 15, 28, 31, 37, 43, 54]. In `OMKP` there are multiple knapsacks with bounded capacity, and the input is a sequence of items, each with an associated weight and value. The goal is to maximize the aggregate value of admitted items such that the sum of the weight of items in each knapsack respects the knapsack's capacity. Upon the arrival of a new item, the online algorithm must decide whether to admit or reject the item; if it admits the item, it should determine in which knapsack the item should be placed. As mentioned in the introduction, our problem is different since we have the item sizes for each dimension as an online input to the problem.

More broadly, our problem is in the category of online admission control problems in multiple dimension. This setting can capture a variety of application domains. Some examples are connection routing and admission control in network [25, 34, 38], cloud computing jobs [30, 35, 53], admission control for electric vehicles at charging stations [2, 43, 46], and QoS buffer management [48]. The similarity between different versions of these problems is that demands dominate the limited resource. In other words, the online algorithm must reject some requests to respect the system's capacity. The online algorithm's decision is mostly based on the current available resource and predictions of future requests. However, those problems are mostly in single dimension setting, while we tackle an online admission control problem in multiple dimensions.

We proceed to review other theoretical problems that are related to OMdKP. A closely related problem is the online bin packing [4, 6, 10, 32], in which n items arriving online should be assigned into a set of available bins such that the lowest possible number of bins is used. Different from OMdKP, there is no admission decision in the bin packing problem; however, opening a new bin or putting the most recent arrived item in the available bins is an online dilemma for an online algorithm. While the classic version of pin packing is single dimension, prior literature has tackled the multidimensional bin packing problem as well [16, 19, 45]. In the multidimensional bin packing problem, the size of items and capacity of bins are represented as a k dimensional vector. In most cases, the capacity of bins in all dimensions is the same, however, the size of a particular item in different dimensions may vary. In other words, bins are hyper-cubes while items are hyper-rectangle. While packing multidimensional items into bins is similar to the OMdKP, the difference between these two problems is clear since there is no admission control in the online bin packing problem. Also, opening a new bin is a flexibility that is not a design space in OMdKP.

We note that recently in literature some generic online resource allocation problems have been studied [5, 7], that capture generic resource allocation and linear packing problems. However, they cannot fully capture the problem of interest in this paper, and hence it becomes infeasible to to compare their algorithms with ours. Specifically, the authors in [5] presents an online algorithm for the online packing problems in which n items arrive sequentially and upon arrival item i, a generic objective function should be optimized respect to a packing constraint. The first important difference is that the underlying online problem in [5] is a linear problem, while OMdKP is an integer problem. The authors in [7] study another generic online allocation problem with the finite horizon. At time $t \leq T$ a new item with non-negative reward function f_t and non-negative resource consumption function b_t arrive, and the algorithm must decide the amount of resource given to this item, x_t. The goal is to maximize the aggregate reward, i.e., $\sum_{t\in[T]} f_t(x_t)$, while respecting the a resource consumption constraint, i.e., $\sum_{t\in[T]} b_t(x_t) \leq \rho T$, where ρ is m dimensional resource constraint vector. In this problem, the resource constraint is a collective constraint, and there is no individual constraint on resources given to a particular item. This is a significant contrast that makes the online resource allocation problem different from OMdKP. Last, they use regret as the performance metric of the proposed algorithms, while we use competitive ratio.

8 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we developed online algorithms for fractional and integral versions of the online multidimensional knapsack problem. Our algorithms are based on carefully designed linear and exponential reservation policies and achieve bounded competitive ratios for both fractional and integral settings. By characterizing a lower bound for the competitive ratio of any online algorithm solving the problem, we also showed that the competitive ratios of our exponential reservation policies for problem instance with small item weights matches the lower bounds up to a constant factor. Later, we numerically verified the performance of the proposed algorithms under a variety of experimental scenarios and compared them with multiple baseline algorithms.

An interesting future work is to design online algorithms that relax the need for a bound on item size for the integral model. One possible approach to relax this assumption is to develop a proper randomized strategy that outputs an integral decision from our competitive fractional algorithm. Another future direction is to relax the bounded assumptions on the values of items. An initial idea is to extend the existing algorithms that relax this assumptions for one-way trading problem [18]. Another promising future direction is on integrating the predictive models into the algorithms design to improve the empirical performance of the competitive algorithms. The motivation is clear from our empirical results where in several cases, `LinRP`, the algorithm with a weaker theoretical guarantee than the lower bound, performs empirically better than `ExpRP`, the one with a better competitiveness in worst case. This observation shows that developing worst-case optimized algorithms is not sufficient to achieve theoretical and practical efficiency at the same time. We highlight two initial ideas to achieve this goal of the best of both worlds. First, one can systematically integrate the predictions into the design of online algorithms. This could be accomplished by using the new framework of online algorithms with ML advice [33, 36, 40, 41]. In this model, it is assumed that some prediction of future is available in terms of advice from ML models, and the goal is to integrate them into the design of online algorithms while keeping the worst-case competitive ratios. The second approach is motivated from the fact that rather than using hand-crafted worst-case optimized algorithms, practitioners prefer to optimize over a class of algorithms and tune the parameters of these algorithms to find an algorithm with improved performance in practice. To do this, we need to define a class of parametric online algorithms [50], as a meta-algorithm, and run online learning approaches in run time to find the best practical algorithm within them. This idea is applied to the basic online knapsack problem in [50], however, applying this to the multidimensional case calls for new algorithm design.

ACKNOWLEDGMENTS

This research is supported by NSF CAREER 2045641, CNS 2106299, CNS 2102963, CNS 1908298, CNS 1763617, CNS 1901137, the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 (IoBT CRA), and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement W911NF-16-3-0001.

REFERENCES

[1] Bahram Alinia, Mohammad H Hajiesmaili, and Noël Crespi. 2019. Online EV charging scheduling with on-arrival commitment. *IEEE Transactions on Intelligent Transportation Systems* 20, 12 (2019), 4524–4537.

[2] Bahram Alinia, Mohammad H Hajiesmaili, Zachary J Lee, Noel Crespi, and Enrique Mallada. 2020. Online EV scheduling algorithms for adaptive charging networks with global peak constraints. *IEEE Transactions on Sustainable Computing* (2020).

[3] Bahram Alinia, Mohammad Sadegh Talebi, Mohammad H Hajiesmaili, Ali Yekkehkhany, and Noel Crespi. 2018. Competitive online scheduling algorithms with applications in deadline-constrained EV charging. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[4] Yossi Azar, Joan Boyar, Lene M Favrholdt, Kim S Larsen, Morten N Nielsen, and Leah Epstein. 2002. Fair versus unrestricted bin packing. *Algorithmica* 34, 2 (2002), 181–196.

[5] Yossi Azar, Niv Buchbinder, TH Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, et al. 2016. Online algorithms for covering and packing problems with convex objectives. In *Proc. of FOCS*. 148–157.

[6] János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. 2021. A new lower bound for classic online bin packing. *Algorithmica* (2021), 1–16.

[7] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. 2020. The Best of Many Worlds: Dual Mirror Descent for Online Allocation Problems. *arXiv preprint arXiv:2011.10124* (2020).

[8] Allan Borodin and Ran El-Yaniv. 2005. *Online computation and competitive analysis*. Cambridge University Press.

[9] Sylvain Boussier, Michel Vasquez, Yannick Vimont, Saïd Hanafi, and Philippe Michelon. 2010. A multi-level search strategy for the 0–1 Multidimensional Knapsack Problem. *Discrete Applied Mathematics* 158, 2 (2010), 97 – 109. https://doi.org/10.1016/j.dam.2009.08.007

[10] Joan Boyar, Lene M Favrholdt, Kim S Larsen, and Morten N Nielsen. 2001. The competitive ratio for on-line dual bin packing with restricted input sequences. *Nordic Journal of Computing* 8, 4 (2001), 463–472.

[11] Niv Buchbinder and Joseph Naor. 2009. *The design of competitive online algorithms via a primal-dual approach*. Now Publishers Inc.

[12] Niv Buchbinder and Joseph Naor. 2009. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research* 34, 2 (2009), 270–286.

[13] Hans-Joachim Böckenhauer, Dennis Komm, Richard Královič, and Peter Rossmanith. 2014. The online knapsack problem: Advice and randomization. *Theoretical Computer Science* 527 (2014), 61 – 72. https://doi.org/10.1016/j.tcs.2014.01.027

[14] Ying Cao, Bo Sun, and Danny HK Tsang. 2020. Optimal Online Algorithms for One-Way Trading and Online Knapsack Problems: A Unified Competitive Analysis. In *Proc. of IEEE Conference on Decision and Control (CDC)*. 1064–1069.

[15] Deeparnab Chakrabarty, Yunhong Zhou, and Rajan Lukose. 2008. Online knapsack problems. In *Workshop on internet and network economics (WINE)*.

[16] Chandra Chekuri and Sanjeev Khanna. 1999. On multi-dimensional packing problems. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. Citeseer, 185–194.

[17] Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. 2004. The all-or-nothing multicommodity flow problem. In *Proc. of ACM STOC*. 156–165.

[18] Francis YL Chin, Bin Fu, Jiuling Guo, Shuguang Han, Jueliang Hu, Minghui Jiang, Guohui Lin, Hing-Fung Ting, Luping Zhang, Yong Zhang, et al. 2015. Competitive algorithms for unbounded one-way trading. *Theoretical Computer Science* 607 (2015), 35–48.

[19] Henrik I Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. 2017. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review* 24 (2017), 63–79.

[20] Marek Cygan, Łukasz Jeż, and Jiří Sgall. 2016. Online knapsack revisited. *Theory of Computing Systems* 58, 1 (2016), 153–190.

[21] Ran El-Yaniv, Amos Fiat, Richard M Karp, and Gordon Turpin. 2001. Optimal search and one-way trading online algorithms. *Algorithmica* 30, 1 (2001), 101–139.

[22] Arnaud Fréville and SaÏd Hanafi. 2005. The multidimensional 0-1 knapsack problem—bounds and computational aspects. *Annals of Operations Research* 139, 1 (2005), 195.

[23] Arnaud Fréville. 2004. The multidimensional 0–1 knapsack problem: An overview. *European Journal of Operational Research* 155, 1 (2004), 1 – 21. https://doi.org/10.1016/S0377-2217(03)00274-1

[24] Waldo Gálvez, Fabrizio Grandoni, Sandy Heydrich, Salvatore Ingala, Arindam Khan, and Andreas Wiese. 2017. Approximating geometric knapsack via L-packings. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 260–271.

[25] Rainer Gawlick, Anil Kamath, Serge Plotkin, and KG Ramakrishnan. 1995. Routing and Admission Control of Virtual Circuits in General Topology Networks. (1995).

[26] Glauco Estácio Gonçalves, Patricia Takako Endo, Moises Rodrigues, Djamel H Sadok, Judith Kelner, and Calin Curescu. 2020. Resource allocation based on redundancy models for high availability cloud. *Computing* 102, 1 (2020), 43–63.

[27] Naman Govil, Rahul Shrestha, and Shubhajit Roy Chowdhury. 2017. PGMA: an algorithmic approach for multi-objective hardware software partitioning. *Microprocessors and Microsystems* 54 (2017), 83–96.

[28] Xin Han and Kazuhisa Makino. 2010. Online removable knapsack with limited cuts. *Theoretical Computer Science* 411, 44 (2010), 3956 – 3964. https://doi.org/10.1016/j.tcs.2010.08.009

[29] Mhand Hifi, Mustapha Michrafy, and Abdelkader Sbihi. 2004. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *Journal of the Operational Research Society* 55, 12 (2004), 1323–1332.

[30] Dinh Thai Hoang, Dusit Niyato, and Ping Wang. 2012. Optimal admission control policy for mobile cloud computing hotspot with cloudlet. In *2012 IEEE wireless communications and networking conference (WCNC)*. IEEE, 3145–3149.

[31] Kazuo Iwama and Shiro Taketomi. 2002. Removable online knapsack problems. In *International Colloquium on Automata, Languages, and Programming*. Springer, 293–305.

[32] David S Johnson. 1973. *Near-optimal bin packing algorithms*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[33] Russell Lee, Jessica Maghakian, Mohammad Hajiesmaili, Jian Li, Ramesh Sitaraman, and Zhenhua Liu. 2021. Online Peak-Aware Energy Scheduling with Untrusted Advice. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems (ACM eEnergy)*. 17.

[34] Jérémie Leguay, Lorenzo Maggi, Moez Draief, Stefano Paris, and Symeon Chouvardas. 2016. Admission control with online algorithms in SDN. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 718–721.

[35] Brendan Lucier, Ishai Menache, Joseph Naor, and Jonathan Yaniv. 2013. Efficient online scheduling for deadline-sensitive jobs. In *Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*. 305–314.

[36] Thodoris Lykouris and Sergei Vassilvtiskii. 2018. Competitive caching with machine learned advice. In *International Conference on Machine Learning*. PMLR, 3296–3305.

[37] Alberto Marchetti-Spaccamela and Carlo Vercellis. 1995. Stochastic on-line knapsack problems. *Mathematical Programming* 68, 1 (1995), 73–104.

[38] S. Paris, J. Leguay, L. Maggi, M. Draief, and S. Chouvardas. 2016. Online experts for admission control in SDN. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 1003–1004.

[39] Jakob Puchinger, Günther R Raidl, and Ulrich Pferschy. 2010. The multidimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing* 22, 2 (2010), 250–265.

[40] Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems (NeurIPS* 31 (2018), 9661–9670.

[41] Bo Sun, Russell Lee, Mohammad H. Hajiesmaili, Adam Wierman, and Danny Tsang. 2021. Pareto-Optimal Learning-Augmented Algorithms for Online Conversion Problems. *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[42] Bo Sun, Tongxin Li, Steven H Low, and Danny HK Tsang. 2020. ORC: An Online Competitive Algorithm for Recommendation and Charging Schedule in Electric Vehicle Charging Network. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*. 144–155.

[43] Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny HK Tsang. 2020. Competitive Algorithms for the Online Multiple Knapsack Problem with Application to Electric Vehicle Charging. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3 (2020), 1–32.

[44] Xiaoqi Tan, Bo Sun, Alberto Leon-Garcia, Yuan Wu, and Danny HK Tsang. 2020. Mechanism design for online resource allocation: A unified approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 2 (2020), 1–46.

[45] Alessio Trivella and David Pisinger. 2016. The load-balanced multi-dimensional bin-packing problem. *Computers & Operations Research* 74 (2016), 152–164.

[46] Nathaniel Tucker and Mahnoosh Alizadeh. 2019. An online admission control mechanism for electric vehicles at public parking infrastructures. *IEEE Transactions on Smart Grid* 11, 1 (2019), 161–170.

[47] M Jalali Varnamkhasti. 2012. Overview of the algorithms for solving the multidimensional knapsack problems. *Advanced Studies in Biology* 4, 1 (2012), 37–47.

[48] Lin Yang, Wing Shing Wong, and Mohammad H Hajiesmaili. 2017. An optimal randomized online algorithm for QoS buffer management. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 2 (2017), 1–26.

[49] Quan Yuan and Zhixin Yang. 2013. An Improved Weight-coded Evolutionary Algorithm for the Multidimensional Knapsack Problem. *arXiv preprint arXiv:1302.5374* (2013).

[50] Ali Zeynali, Bo Sun, Mohammad Hajiesmaili, and Adam Wierman. 2021. Data-driven Competitive Algorithms for Online Knapsack and Set Cover. In *Proc. of AAAI*.

[51] Xiaoxi Zhang, Zhiyi Huang, Chuan Wu, Zongpeng Li, and Francis CM Lau. 2015. Online auctions in IaaS clouds: Welfare and profit maximization with server costs. In *Proc. of ACM SIGMETRICS*. 3–15.

[52] Zijun Zhang, Zongpeng Li, and Chuan Wu. 2017. Optimal posted prices for online cloud resource allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–26.

[53] Zizhan Zheng and Ness B Shroff. 2016. Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[54] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. 2008. Budget constrained bidding in keyword auctions and online knapsack problems. In *International Workshop on Internet and Network Economics*. Springer, 566–576.

A SUPPLEMENTARY PROOFS

A.1 Proof of Theorem 1

Consider a simple instance where the system execute a FCFS strategy to admit items. The adversary can exhaust the space in the first dimension by repeatedly presenting the items with value 1 and the following weights to the FCFS algorithm for n times.

$$\left[\frac{C_1}{n}, 0, \ldots, 0\right].$$

Afterwards, the adversary presents the items with value θ and the following weights to the FCFS strategy for another n times.

$$\left[\frac{C_1}{n}, \frac{C_2}{n}, \ldots, \frac{C_m}{n}\right].$$

The FCFS strategy can only admit the first n items and will miss the rest, since it already uses up the space in the first dimension to admit the first n items. Thus, the aggregate value of items admitted by FCFS is C_1, while that earned by the optimal algorithm, which admits the last n items, is θC. In this way, we show that the competitive ratio of the FCFS strategy is at least $(\theta C)/C_1 = \theta\alpha$, completing the proof.

A.2 Proof of the Last Inequality in Equation (11)

We use Figure 6 to facilitate our proof. Specifically, $\sum_{i=1}^{n}\left\lfloor \frac{u_{i-1,j'}}{C_{j'}}\sqrt{\theta m}\right\rfloor \left(u_{i,j'} - u_{i-1,j'}\right)$ can be seen as an approximation of the integral of the step function $\left\lfloor \frac{u_{i-1,j'}}{C_{j'}}\sqrt{\theta m}\right\rfloor$ with step length being $\frac{C_{j'}}{\sqrt{\theta m}}$, and be visualized by the colored area in Figure 6. By calculating a lower bound for the size of the colored area, we prove the last inequality in Equation (11). We note that the methodology used above is also applied to other proofs in the paper, e.g., the one for Equation (13).

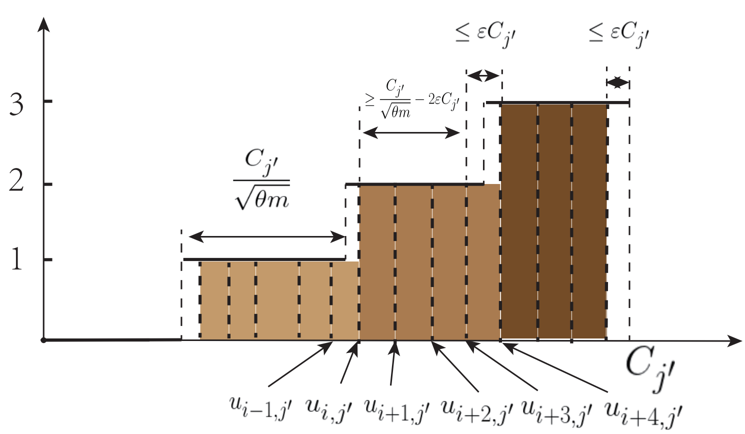

Fig. 6. Visualized proof for Equation (11)

A.3 A Proof for the Lower Bound Result in Theorem 2

The core idea to prove the lower bound of competitive ratio is to construct two adversaries under which the competitive ratio of any online algorithm is $\Omega(\log \alpha)$ and $\Omega(\log \theta)$, respectively. Accordingly, the proof contains the following two steps.

Step 1: Prove lower bound $\Omega(\log \alpha)$.

To prove the lower bound, we define Adversary 1 which generates a series of inputs as follows.

Generally, Adversary 1 runs multiple rounds, at each which it repeatedly presents a particular type of items of unit density to the investigated online algorithm. During the l-th round, l =

$1, 2, \ldots, C/C_1$,[1] we repeatedly present a job satisfying $w_{i,1} = \delta \log^{-1} \frac{C}{C_1}$ and $w_{i,j} = (l-1)C_1 C_j/(C - C_1)$, $j = 2, 3, \ldots, m$, to the investigated online algorithm where δ is a small positive. The adversary can end at anytime. Specifically, Input-l refers to the input which ends after the l-th round.

Let y_l be the number of admitted items by the online algorithm at the l-th round. To guarantee a competitive ratio less than $\log \frac{C}{C_1}$ for the l-th input, $l = 1, 2 \ldots, C/C_1$, we have

$$\sum_{l'=1}^{l} l' y_{l'} > lC_1/\delta. \tag{15}$$

Otherwise, the algorithm will have a competitive ratio that is at least $\log \frac{C}{C_1}$, because

$$\frac{lC_1}{\sum_{l'=1}^{l} l' y_{l'} \delta \log^{-1} \frac{C}{C_1}} \geq \frac{lC_1}{\delta \log^{-1} \frac{C}{C_1} \frac{lC_1}{\delta}} = \log \frac{C}{C_1},$$

where lC_1 corresponds to the cumulative values received by the optimal algorithm and the term $\sum_{l'=1}^{l} l' y_{l'} \delta \log^{-1} \frac{C}{C_1}$ corresponds to that of the online algorithm. Moreover, y_l should satisfy the capacity constraint, i.e.,

$$\sum_{l=1}^{C/C_1} y_l \delta \log^{-1} \frac{C}{C_1} \leq C_1, \tag{16}$$

where the left hand side of the above equation is cumulative weighs in dimension 1 by the online algorithm. Then, we can prove our result by showing that there are no feasible solutions for y_l, $l = 1, 2, 3, \ldots, C/C_1$ that simultaneously satisfy Equations (15) and (16) (see Lemma 1 in appendix). Thus, the competitive ratio of the online algorithm is always larger than or equal to $\log(C/C_1)$.

Step 2: Prove a lower bound, $\log \frac{\theta}{1+\delta} + 1$ for any $\delta > 0$.

To prove the above lower bound, we define Adversary 2 which runs L rounds. At each round, Adversary 2 repeatedly presents the same type of items which only demands on resource 1. During the l-th round, $l = 1, 2, \ldots, L$, we repeatedly present a job satisfying $w_{i,1} = C_1$ and $v_i = C_1(1+(l-1)\delta)$, to the investigated online algorithm where $\delta = (\theta - 1)/(L - 1)$. Also, the adversary can end at anytime. Specifically, Input-l refers to the input which ends after the l-th round.

Let y_l be the number of admitted jobs by a deterministic online algorithm at the l-th round, or be the expectation if the algorithm is randomized. Up to the l-th round, the value earned by the optimal algorithm is $(1 + l\delta)C_1$, which is achieved by admitting the item at the l-th round. To guarantee a competitive ratio of $\log \frac{\theta}{1+\delta} + 1$ for the l-th input, $l = 1, 2 \ldots, L$, there is

$$\sum_{l'=1}^{l} (1 + (l'-1)\delta)\, y_{l'} C_1 \geq (1 + (l-1)\sigma) C_1 \left(\log \frac{\theta}{1+\delta} + 1\right)^{-1}. \tag{17}$$

Moreover, y_l should satisfy the capacity constraint, i.e.,

$$\sum_{l=1}^{L} y_l C_1 \leq C_1. \tag{18}$$

Then, we will show that there are no feasible solutions for y_l, $l = 1, 2, 3, \ldots, L$ that simultaneously satisfy Equations (17) and (18).

---

[1] Without loss of generality, we assume C/C_1 to be an integer.

Let $Q_0 := 0$ and

$$Q_l := \sum_{l'=1}^{l} (1 + (l' - 1)\sigma)\, y_{l'} C_1,\ l = 1, 2, \ldots, L.$$

We have

$$\begin{aligned}
\sum_{l=1}^{L} y_l C_1 &= \sum_{l=1}^{L} (Q_l - Q_{l-1}) \frac{1}{1 + (l-1)\delta} \\
&= \sum_{l=1}^{L-1} Q_l \left(\frac{1}{1 + (l-1)\delta} - \frac{1}{1 + l\delta} \right) + \frac{Q_L}{\theta} \\
&\geq \sum_{l=1}^{L-1} (1 + (l-1)\delta) \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} \left(\frac{1}{1 + (l-1)\delta} - \frac{1}{1 + l\delta} \right) C_1 + C_1 \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} \\
&= \sum_{l=1}^{L-1} \frac{\delta}{1 + l\delta} C \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} + C_1 \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} \\
&> C_1 \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} \int_{1+\delta}^{\theta} \frac{1}{x} dx + C_1 \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} \\
&= C_1 \log \frac{\theta}{1+\delta} \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} + C_1 \left(\log \frac{\theta}{1+\delta} + 1 \right)^{-1} = C_1,
\end{aligned}$$

where the first inequality is from Equation (17) and the definition of Q_l. The above result contradicts Equation (18). Note that δ can be arbitrarily small when L is large enough. This proves a lower bound $\log \frac{\theta}{1+\delta} + 1$ for any $\delta > 0$.

Combining the above two steps, we can lower bound the competitive ratio of any algorithm, $\mathtt{A}$, either stochastic or deterministic by

$$\mathtt{CR}(\mathtt{A}) \geq \max \{\Omega(\log \alpha), \Omega(\log \theta)\} \geq \frac{1}{2}\Omega(\log \alpha) + \frac{1}{2}\Omega(\log \theta) = \Omega(\log \theta\alpha).$$

This completes the proof.

Lemma 1. *Assume C/C_1 is an integer. There is no positive y_l, $l = 1, 2, \ldots, \frac{C}{C_1}$, satisfying*

$$\sum_{l'=1}^{l} l' y_{l'} > \frac{l C_1}{\delta},$$

and

$$\sum_{l=1}^{C/C_1} y_l \delta \log^{-1} \frac{C}{C_1} \leq C_1.$$

Proof. We prove the lemma by contradiction. We assume there exist y_l, $l = 1, 2, \ldots, \frac{C}{C_1}$, satisfying the above two equations. Let $A_0 := 0$ and $A_l := \sum_{l'=1}^{l} l' y_{l'}$ for $l = 1, 2, \ldots, \frac{C}{C_1}$. Then, we have $A_l > \frac{lC_1}{\delta}$. There is

$$
\begin{aligned}
\sum_{l=1}^{C/C_1} y_l \delta \log^{-1} \frac{C}{C_1} &= \sum_{l=1}^{C/C_1} (A_l - A_{l-1}) \frac{1}{l} \delta \log^{-1} \frac{C}{C_1} \\
&= \sum_{l=1}^{C/C_1-1} \left(\frac{1}{l} - \frac{1}{l+1} \right) A_l \delta \log^{-1} \frac{C}{C_1} + \frac{A_{C/C_1}}{C/C_1} \delta \log^{-1} \frac{C}{C_1} \\
&> \sum_{l=1}^{C/C_1-1} \left(\frac{1}{l} - \frac{1}{l+1} \right) l C_1 \log^{-1} \frac{C}{C_1} + C_1 \log^{-1} \frac{C}{C_1} \\
&= \sum_{l=1}^{C/C_1-1} \frac{1}{l+1} \log^{-1} C_1 \frac{C}{C_1} + C_1 \log^{-1} \frac{C}{C_1} > C_1,
\end{aligned}
$$

where the last inequality uses the fact that $1 + 1/2 + 1/3 + \cdots + 1/(C/C_1) > \log \frac{C}{C_1}$. This contradicts the assumption and completes the proof. □

B PROOF FOR THE FRACTIONAL RESULT IN THEOREM 5

Let $\mathbf{Z} = [z_1, z_2, \dots, z_m]$ be the ending state of the system executing the proposed ExpRP-F algorithm. Let $\mathcal{J}_l$, $l = 0, 1, 2, \dots, \lfloor \log(\theta\alpha) \rfloor$, be the set of resources satisfying $z_l \geq l$. Note that, at each round i, the ExpRP algorithm runs multiple rounds to determine x_i. Specifically, at each round, the ExpRP algorithm will add a positive value y. Let $y_{i,r}$, $r = 1, 2, \dots, n_i$ be the value of y generated by ExpRP-F at the r-th round for item i. Obviously, $x_i = \sum_r^{n_i} y_{i,r}$. Accordingly, we define $u_{i,j,0}$ as $u_{i-1,j}$, and $u_{i,j,r}$ as $u_{i-1,j} + \sum_{k=1}^{r} y_{i,k} w_{i,j}$, $r = 1, 2, \dots, n_i$.

The proof is executed case by case.

Case 1: $u_{n,j} < C_j$, for any $j \in [m]$.

First, we provide a lower bound for cumulative values of admitted items by the ExpRP algorithm, which is denoted by $V_{\text{ExpRP-F}}$.

$$
\begin{aligned}
V_{\text{ExpRP-F}} &= \sum_{i=1}^{n} \sum_{r=1}^{n_i} y_{i,r} \sum_{j \in [m]} w_{i,j} \\
&\geq \sum_{i=1}^{n} \sum_{r=1}^{n_i} \sum_{j \in [m]} \left(2^{\left\lfloor \frac{u_{i,r-1,j}}{C_j} \log(\theta\alpha_j) \right\rfloor} - 1 \right) y_{i,r} w_{i,j} \\
&\geq \sum_{i=1}^{n} \sum_{r=1}^{n_i} \sum_{j \in [m]} \left(2^{\left\lfloor \frac{u_{i,r-1,j}}{C_j} \log(\theta\alpha_j) \right\rfloor} - 1 \right) (u_{i,r,j} - u_{i,r-1,j}) \\
&\geq \sum_{j \in \mathcal{J}_1} \sum_{l=1}^{z_j - 1} C_j \log^{-1} (\theta\alpha_j) \left(2^l - 1 \right) \\
&= \sum_{j \in \mathcal{J}_1} C_j \log^{-1} (\theta\alpha_j) \left(2 \left(2^{z_j - 1} - 1 \right) - z_j + 1 \right) \\
&= \sum_{j \in \mathcal{J}_1} C_j \log^{-1} (\theta\alpha_j) \left(2^{z_j} - z_j - 1 \right).
\end{aligned}
$$

In addition, we have

$$V_{\mathsf{ExpRP\text{-}F}} \geq \sum_{j \in \mathcal{J}_1} C_j \log^{-1}\left(\theta \alpha_j\right).$$

Combining the above two equations yields

$$\begin{aligned} V_{\mathsf{ExpRP\text{-}F}} \geq & \max \left\{ \sum_{j \in \mathcal{J}_1} C_j \log^{-1}\left(\theta \alpha_j\right), \sum_{j \in \mathcal{J}_1} C_j \log^{-1}\left(\theta \alpha_j\right) \left(2^{z_j} - z_j - 1\right) \right\} \\ \geq & \beta \sum_{j \in \mathcal{J}_1} C_j \log^{-1}\left(\theta \alpha_j\right) + (1-\beta) \sum_{j \in \mathcal{J}_1} C_j \log^{-1}\left(\theta \alpha_j\right) \left(2^{z_j} - z_j - 1\right), \end{aligned} \tag{19}$$

for any $\beta \in [0, 1]$. Next, we prove an upper bound for the aggregated value of admitted items by any algorithm.

An item i with weight vector $\boldsymbol{w}_i$ is called the Type-I item if the following equation holds.

$$v_i < \sum_{j \in [m]} \left(2^{z_j} - 1\right) w_{i,j}.$$

Otherwise, an item is called the Type-II item. Obviously, an item is either a Type-I item or a Type-II item. In addition, all of Type-II items are accepted by the ExpRP-F algorithm since each of them satisfies

$$v_i \geq \sum_{j \in [m]} \left(2^{z_j} - 1\right) w_{i,j} \geq \sum_{j \in [m]} \left(2^{z_{i-1,j}} - 1\right) w_{i,j}.$$

Next, we will provide an upper bound for the total values of admitted Type-I items by any online algorithm.

Let i_k, $k = 1, 2, \ldots, n_1$, be the indices of Type-I items and x_{i_k}, $k = 1, 2, \ldots, n_1$, be the decisions correspondingly. Then, by the definition of the Type-I item, we have that for any i_k, $k = 1, 2, \ldots, n_1$, there is

$$x_{i_k} v_{i_k} < x_{i_k} \sum_{j=1}^{m} \left(2^{z_j} - 1\right) w_{i_k,j}.$$

It follows from the above equation that

$$\begin{aligned} \sum_{k=1}^{n_1} x_{i_k} v_{i_k} &< \sum_{k=1}^{n_1} x_{i_k} \sum_{j=1}^{m} \left(2^{z_j} - 1\right) w_{i_k,j} \\ &= \sum_{k=1}^{n_1} \sum_{j=1}^{m} \left(2^{z_j} - 1\right) x_{i_k} w_{i_k,j} \\ &= \sum_{j=1}^{m} \left(2^{z_j} - 1\right) \sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \\ &\leq \sum_{j=1}^{m} \left(2^{z_j} - 1\right) C_j, \end{aligned}$$

where the last inequality uses the fact that $\sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \leq C_j$, $j = 1, 2, \ldots, m$. By the above equation, we upper bound the total amount of values of Type-I items that are admitted by any algorithm, which is at most $\sum_{j=1}^{m} \left(2^{z_j} - 1\right) C_j$. In addition, the total amount of values of Type-II items is not larger than the that of admitted items by the ExpRP-F algorithm. Thus, the total amount of values of items admitted by any algorithm is then upper bounded by

$$\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j + V_{\texttt{ExpRP-F}}.$$

Combined with the results in Equation (19) with $\beta = 1/4$, the competitive ratio of the ExpRP-F algorithm satisfies

$$\begin{aligned}
\texttt{CR}(\texttt{ExpRP-F}) \leq & \frac{\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j + V_{\text{on}}}{V_{\text{on}}} \\
= & \frac{\sum_{j=1}^{m} (2^{z_j} - 1)\, C_j}{V_{\text{on}}} + 1 \\
\leq & \frac{\sum_{j\in\mathcal{J}_1} C_j + \sum_{j\in\mathcal{J}_2} (2^{z_j} - 1)\, C_j}{\frac{1}{4}\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right) + \frac{3}{4}\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right)\left(2^{z_j} - z_j - 1\right)} + 1 \\
\leq & \max\left\{4\frac{\sum_{j\in\mathcal{J}_1} C_j}{\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta\alpha_j\right)}, \frac{4\sum_{j\in\mathcal{J}_2} (2^{z_j} - 1)\, C_j}{3\sum_{j\in\mathcal{J}} C_j \log^{-1}\left(\theta\alpha_j\right)\left(2^{z_j} - z_j - 1\right)}\right\} + 1 \\
\leq & 4\log\left(\theta\alpha\right) + 1,
\end{aligned}$$

where the last inequality uses the fact that

$$\frac{2^{z_j} - 1}{2^{z_j} - z_j - 1} \leq 3, \text{ for any } j \in \mathcal{J}_2.$$

Case 2: $u_{n,j} = C_j$, for some $j \in [m]$.

Without loss of generality, we assume that dimension j' satisfies $u_{n,j'} = C_{j'}$. Assume $\theta\alpha_{j'} \geq 8$. Otherwise, there is

$$\texttt{CR}(\texttt{ExpRP-F}) \leq \frac{\theta C}{C_{j'}} \leq 8.$$

By the rules of the ExpRP algorithm, we have

$$
\begin{aligned}
V_{\mathrm{ExpRP\text{-}F}} &= \sum_{i=1}^{n}\sum_{r=1}^{n_i} y_{i,r} v_i \\
&\geq \sum_{i=1}^{n}\sum_{r=1}^{n_i}\sum_{j=1}^{m}\left(2^{\left\lfloor \frac{u_{i,r-1,j}}{C_j}\log\left(\theta\alpha_j\right)\right\rfloor}-1\right) y_{i,r} w_{i,j} \\
&\geq \sum_{i=1}^{n}\sum_{r=1}^{n_i}\left(2^{\left\lfloor \frac{u_{i,r-1,j'}}{C_{j'}}\log\left(\theta\alpha_{j'}\right)\right\rfloor}-1\right) y_{i,r} w_{i,j'} \\
&= \sum_{i=1}^{n}\sum_{r=1}^{n_i}\left(2^{\left\lfloor \frac{u_{i,r-1,j'}}{C_{j'}}\log\left(\theta\alpha_{j'}\right)\right\rfloor}-1\right)\left(u_{i,r,j'}-u_{i,r-1,j'}\right) \\
&\geq \sum_{l=1}^{\left\lfloor \log\left(\theta\alpha_{j'}\right)\right\rfloor}\left(2^{l}-1\right) C_{j'}\log^{-1}\left(\theta\alpha_{j'}\right) \\
&= \left(2\left(2^{\left\lfloor \log\left(\theta\alpha_{j'}\right)\right\rfloor}-1\right)-\log\left(\theta\alpha_{j'}\right)+1\right) C_{j'}\log^{-1}\left(\theta\alpha_{j'}\right) \\
&\geq \left(2^{\log\left(\theta\alpha_{j'}\right)}-\log\left(\theta\alpha_{j'}\right)-1\right) C_{j'}\log^{-1}\left(\theta\alpha_{j'}\right) \\
&= \left(\theta\alpha_{j'}-\log\left(\theta\alpha_{j'}\right)-1\right) C_{j'}\log^{-1}\left(\theta\alpha_{j'}\right) \geq \frac{1}{2}\theta C\log^{-1}\left(\theta\alpha_{j'}\right),
\end{aligned}
$$

where the last inequality uses the fact that $\theta\alpha_{j'} \geq 8$.

Thus,

$$
\mathrm{CR(ExpRP\text{-}F)} \leq \frac{\theta C}{\frac{1}{2}\theta C\log^{-1}\left(\theta\alpha_{j'}\right)} = 2\log\left(\theta\alpha_{j'}\right) = 2\log\left(\theta\alpha\right).
$$

Concluding the above two cases yields

$$
\mathrm{CR(ExpRP\text{-}F)} \leq \max\left\{8, 4\log\theta\alpha\right\} + 1.
$$

Received August 2021; revised October 2021; accepted November 2021