Optimal Competitive Algorithms for Online Multidimensional Knapsack Problems

ANONYMOUS AUTHOR(S)

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 with a different capacity. Then, n items with different scalar values and m-dimensional sizes arrive in an online manner and the goal is to admit or decline items upon their arrival such that the 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 first tackle a basic version of OMdKP with unit densities, in which the unit value of each item is one. We develop two algorithms for unit-density 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 $O(\sqrt{\alpha})$ and $O(\log \alpha)$, where α is the ratio between the aggregate knapsack capacity and the minimum capacity over a single dimension. Then, we tackle the general version of OMdKP and develop linear and exponential reservation algorithms with the competitive ratios of $O(\sqrt{\theta\alpha})$ and $O(\log(\theta\alpha))$, respectively, where θ is the ratio between the maximum and minimum item unit-values. We also characterize a lower bound for competitive ratio of any online algorithm solving OMdKP and show that the competitive ratio of our exponential algorithms matches the lower bound up to a constant factor.

ACM Reference Format:
Anonymous Author(s). 2021. Optimal Competitive Algorithms for Online Multidimensional Knapsack Problems. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 1 (February 2021), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The online knapsack problem [12] (OKP) is a classical online optimization problem that has several applications in a variety of domains such as cloud and edge computing [27, 42], online admission control [9], online routing of virtual switches [30], and control of distributed energy resources in smart grids [1–3, 32]. 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. This problem has been tackled using the competitive online algorithm framework [7], and there are algorithms [10, 12, 43] that achieve the competitive ratio of $O(\log \theta)$ for the basic version, where θ is the value fluctuation ratio parameter denoting the ratio between the most and the least valuable items. Further, is has been shown that the lower bound on the competitive ratio is $O(\log \theta)$ [43], hence the algorithms in [41, 43] are optimal since their competitive ratio is tight.

Recently, the basic OKP has been extended to more practical extensions to better capture properties of real-world applications. In [10], an extended version of OKP, the online multiple knapsack problem, has been considered in which there are multiple knapsacks with different capacities, and divisible

2476-1249/2021/2-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

items could be packed into the knapsacks. Hence, the question becomes whether to admit or decline, and then if admitted, how to pack the item into the existing knapsacks with available capacities. In another work [34], 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, [41] extended OKP to a time-expanded version. In [40], the basic setting is extended to knapsacks with packing costs. We will review the related work 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 as an m-dimensional vector, and the size (or weight, used interchangeably) of online items are represented in m dimensions. The goal of an online decision maker is to pack the most valuable items such that the capacity of the knapsack in each dimension is respected. Note that this problem differs from the online multiple knapsack problem in which fixed size items can be packed into one or multiple knapsacks. In other words, in the online multiple knapsack problem, the *admission* and *allocation* is the design space, however in OMdKP, the admission is the only decision, and allocation over dimensions is an input to the decision maker.

The multidimensional knapsack problem (MdKP) is of significant practical relevance since it applies to scenarios where there are different types of resources and demands of items (e.g., jobs or virtual machines) for each resource is given as the input. Hence, MdKP has been extensively studied in the literature [16, 17, 31, 36, 38]. However, all these works tackle the problem in the offline setting. In contrast, we focus on the online version where the goal is to design competitive algorithms for OMdKP.

The high-level idea of our algorithm design follows designing online reservation (a.k.a. threshold) functions for admitting online items. Specifically, the goal in this approach is to design reservation functions that associate an *implicit admission cost* to the knapsack as a function of its utilization. The reservation cost function is increasing, 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 [41, 43] and the multiple knapsack version [33], and it has been shown that the corresponding online algorithms can achieve the best competitive ratios.

Designing the reservation policies for OMdKP is more challenging than the basic OKP and the existing algorithms for OKP are not applicable in this setting. More specifically in OMdKP, the item size is multidimensional and the element-wise size might be unbalanced across different dimensions. On the other hand, there is a single scalar value associated to each item. Hence, items with the same value and aggregate size will be treated the same using the existing algorithms for OKP, while their element-wise sizes might be different, e.g., consider two items with the same values and same aggregate size, one demanding more on a dimension that is highly utilized, while the other item has bigger size for an under-utilized dimension.

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 sizes across different dimensions. The reservation policies are a linear function and an exponential function of knapsack utilization and demonstrate different competitive ratios for the corresponding online algorithms. The main contribution of this paper is characterizing the competitive ratios of the online algorithms with exponential reservation policies as a logarithmic function of problem parameters and showing their optimality since they match a logarithmic lower bound that we characterize for the competitive ratio of any online algorithm providing a feasible solution to OMdKP. The details of the contributions are summarized in the following.

For better illustration of algorithms and results, we first present our results for OMdKP-UD, which is a unit-density version of OMdKP where the value of each item is equal to its aggregate size over dimensions. We develop two algorithms based on linear (LinRP) and exponential (ExpRP) reservation policies and characterize their competitive ratios. Our competitive analysis shows that LinRP achieves a competitive ratio of $O(\sqrt{\alpha})$, where α is the ratio between the aggregate knapsack capacity and the minimum capacity over a single dimension, i.e., $\alpha = \sum_j C_j / \min_j C_j$, where C_j is the capacity of dimension j. The competitive ratio of ExpRP is $O(\log \alpha)$. Then, we characterize a lower bound of $O(\log \alpha)$ for the competitive ratio of any online algorithm solving OMdKP-UD. Hence, ExpRP attains the optimal competitive ratio up to a constant factor.

Second, we extend both algorithms to general OMdKP and develop LinRP-G and ExpRP-G, two online algorithms with a linear function and an exponential function, respectively. Our competitive analysis shows competitive ratios of $O(\sqrt{\theta\alpha})$ and and $O(\log \theta\alpha)$ for LinRP-G and ExpRP-G, respectively, where θ is the ratio between the maximum and minimum unit values of items. Comparing the result with the lower bound shows the competitive ratio of ExpRP-G is optimal, and in a single dimension, our algorithm reduces to prior algorithms for OKP [41, 43]. Last, we extend the proposed exponential algorithm and competitive analysis to the fractional version of OMdKP.

It is worth mentioning that when the dimension size is large or dimensions are highly heterogeneous in capacity, the competitive ratio of exponential policies dominates those of linear policies for both OMdKP-UD and OMdKP. However, in practical scenarios with low dimensions, the linear algorithms may outperform the exponential ones; hence both algorithms are practically relevant. In Section 6, we numerically compare the empirical competitive ratios of the proposed algorithms, and our numerical observations verify the above intuition.

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 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 [5, 19], all-or-nothing flow maximization [13], and more. In the following, we formally introduce OMdKP.

Problem Statement. Consider a knapsack whose capacities along m dimensions is represented by vector $\mathbf{C} = [C_1, \dots, C_j, \dots, C_m]$, where C_j represents the capacity of dimension $j \in [m] = \{1, \dots, m\}$. Without loss of generality, we assume, $C_1 \leq C_2 \leq \dots \leq C_m$. Items arrive in an online fashion, each with a different value and size. Specifically, in round $i \in [n] = \{1, \dots, n\}$, item i arrives with value $v_i \geq 0$, and a weight vector $\mathbf{w_i} = [w_{i,1}, \dots, w_{i,j}, \dots, w_{i,m}]$, 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 MdKP can be formulated as

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

where x_i's are the optimization variables. 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 of MdKP. 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/non-stochastic model. We present our 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.

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 $\boldsymbol{u}_i = [u_{i,1}, \dots, u_{i,j}, \dots, u_{i,m}]$ be the knapsack utilization after making a decision to admit item i or not, where $0 \le u_{i,j} \le C_j$ 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 $p_i \in [1, \theta], \forall i \in [n]$, where $\theta \ge 1$ is an upper bound on the unit value of each item. Parameter θ plays a critical role in the competitive analysis of the proposed algorithms. Note that in Equation (2), the unit value is defined as the ratio between the item value and the aggregate size of the item over all m dimensions. However, this definition and our algorithms can be extended to account for weighted aggregate size over the dimensions as we formally explain in Remark 4 after Theorem 4.

In our algorithms, we assume that normalized weights of items are much smaller than the capacity, i.e., $w_{i,j}/C_j \le \varepsilon \ll 1, \forall i, j$, 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 [33, 39]. We present our results by explicit characterization of valid regions for ε. Also, in Section 5, we relax this assumption for our algorithms for the fractional model.

In the algorithm design, we first focus on OMdKP-UD that is OMdKP with *unit-density*. In OMdKP-UD, we assume $p_i = 1, \forall i \in [n]$. In other words, item values are proportional to their sizes and the larger the aggregate size of an item, the higher its value is. In OMdKP-UD, the goal is to maximize the size of admitted items, or equivalently, maximize the utilization of the knapsack. We note that studying the unit-density version of the knapsack is common in prior work [6, 14, 21, 28].

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 [7] with *competitive ratio* as the performance metric. Specifically, for an online algorithm A, the competitive ratio is

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

where $\omega \in \Omega$ denotes a feasible instance to OMdKP and Ω is the set of all feasible instances to OMdKP. Also, $\mathsf{OPT}(\omega)$ is the offline optimum under instance ω, and $\mathsf{A}(\omega)$ is the profit obtained by executing the online algorithm A over instance ω. We present our algorithms for OMdKP-UD in Section 3, for OMdKP in Section 4, and the fractional algorithms in Section 5.

3 ONLINE ALGORITHMS FOR OMDKP-UD

In Section 3.1, we develop competitive online algorithms for OMdKP-UD, in which values of items are the sums of their weights, i.e., $v_i = \sum_{j=1}^{m} w_{i,j}$. Specifically, we propose the LinRP and ExpRP reservation policies and analyze their performance in Section 3.2 and show that they achieve $O(\sqrt{\alpha})$

and $O(\log \alpha)$ competitive ratios, respectively. We also characterize a lower bound for competitive ratios for OMdKP-UD, and show that the competitive ratio of ExpRP matches this lower bound.

3.1 Online Algorithms

To motivate algorithm design for OMdKP-UD, we first analyze the competitiveness of the First-come First-serve (FCFS) strategy and show FCFS is $O(\alpha)$-competitive, hence, we need to design better algorithms with better competitive ratios.

3.1.1 Warm-up: First-Come First-Serve: An $O(\alpha)$-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 α-competitive.

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

The above result shows that FCFS is oblivious to the residual capacity of individual dimensions, which leads to a large 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 (scarcity). Based on this high-level intuition, in the following, we introduce two policies that respectively associate linear and exponential scarcity factors 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 admits an incoming item only when its value is larger than or equal to the current admission cost.

*3.1.2 The Linear Reservation Policy (*LinRP*).* We first introduce a $\sqrt{2\alpha}$-competitive algorithm, called as the Linear Reservation Policy (LinRP). As compared to the naive FCFS strategy, LinRP dynamically adjusts a threshold to admit an item based on the scarcity across each dimension. In doing so, the LinRP algorithm reserves scarce space for high-valued items. Details of the LinRP algorithm are summarized in Algorithm 1.

Algorithm 1 The LinRP Algorithm, upon arrival of item i

1: **Discretization:** Discretize the system state by $z_{i,j}$, $j \in [m]$ defined as

$$z_{i,j} = \left\lfloor \frac{u_{i,j}}{C_j}\sqrt{m} \right\rfloor, j \in [m]. \tag{4}$$

2: **Decision Making:** Admit item i if there is enough space to admit, i.e., $w_{i,j} \le C_j - u_{i-1,j}, \forall j \in [m]$, and the following inequality hold

$$v_i \ge \max_{j \in [m]} z_{i-1,j} \sqrt{\frac{2C}{mC_j}} w_{i,j}. \tag{5}$$

In Equation (5), the left hand side is the value of the incoming item i, and $z_{i-1,j}\sqrt{C/(mC_j)}$ 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 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 the maximum evaluated cost in all dimensions. 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.

3.1.3 The Exponential Reservation Policy (`ExpRP`*).* Now, we proceed to introduce `ExpRP` which uses an exponential reservation function for evaluating the admission cost. The details of the `ExpRP` algorithm are summarized in Algorithm 2.

Algorithm 2 The `ExpRP` Algorithm, upon arrival of item i

1: **Discretization:** Discretize the system state by $z_{i,j}$, $j \in [m]$ defined as

$$z_{i,j} = \left\lfloor \frac{u_{i,j}}{C_j} \log \frac{C}{C_j} \right\rfloor, j \in [m]. \tag{6}$$

2: **Decision Making:** Admit item i, if there is enough space to admit it, i.e., $w_{i,j} \leq C_j - u_{i-1,j}, \forall j \in [m]$, and the following inequality hold

$$v_i \geq \sum_{j \in [m]} (2^{z_{i-1,j}} - 1)\, w_{i,j}. \tag{7}$$

Given enough available space for admission, `ExpRP` makes an admission decision based on Equation (7), 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.

3.2 Competitive Analysis for OMdKP-UD

We now present our main competitive results for both `LinRP` and `ExpRP`.

3.2.1 Main Results. Recall that ε serves as an upper bound of ratios between single-dimension size of items and the corresponding capacity (Equation (3)).

Theorem 2. *(Lower Bound on Competitive Ratio) The competitive ratio of any online algorithm providing a feasible solution to* `OMdKP-UD` *is at least* $\log \alpha$.

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

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

Remarks. (1) When $\varepsilon \to 0$, representing the case with arbitrarily small item sizes, we have

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

(2) Further, with unit dimension capacities, i.e., $C_1 = C_2 = \cdots = C_m = 1$ and $\varepsilon \to 0$, we have

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

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

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

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

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

(2) for unit capacities and $\varepsilon < \min\{1/3, 1/(2\log m)\}$, the competitive ratio of ExpRP satisfies

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

(3) When comparing the competitive ratios of LinRP and ExpRP, one finds that ExpRP outperforms LinRP when α is large. However, when α is small, LinRP can outperform ExpRP as we will observe when we numerically compare their performances in Section 6.

(4) As we mentioned in the system model, we consider the aggregate size of an item to be simply the summation of its size 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 replacing C_j in Equations (4) and (6) with $d_j C_j$. Then, similar competitive ratios can be obtained by setting $\alpha = \sum_{j\in[m]} d_j C_j / \min_{j\in[m]} d_j C_j$.

3.2.2 Proofs. We now present the proofs for the above three theorems.

A Proof for Theorem 2. To prove the lower bound, we construct a series of inputs as follows.

Generally, the adversary runs multiple rounds, at each which it repeatedly presents a particular type of items 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 $w_{i,j} = (l-1)C_1C_j/(C-C_1)$, $j = 2, 3, \ldots, m$, 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

$$\sum_{l'=1}^{l} l' y_{l'} > lC_1/\delta. \tag{8}$$

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.,

$$\sum_{l=1}^{C/C_1} y_l \delta \log^{-1}\frac{C}{C_1} \leq C_1, \tag{9}$$

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,

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

$l = 1, 2, 3, \ldots, C/C_1$ that simultaneously satisfy Equations (8) and (9) (see Lemma 1 in appendix). Thus, the competitive ratio of the online algorithm is always larger than or equal to $\log(C/C_1)$. This completes the proof.

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 $\mathcal{J}_l \subset [m]$, $l = \{0, 1, 2, \ldots, \lfloor\sqrt{m}\rfloor\}$, 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 (5) is satisfied. Consider the following constraints for incoming item i

$$\text{constraint} - j : v_i \geq z_j \sqrt{\frac{2C}{mC_j}} w_{i,j}, \; j \in \mathcal{J}_1. \tag{10}$$

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 (10), we upper bound the aggregate value of admitted Type-j items, denoted as V_j, as follows

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

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

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 (10), it will be admitted by `LinRP`. Thus, the aggregate value of the admitted items satisfying all the constraints in Equation (10), 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 $V_{\texttt{LinRP}} \geq \sum_{j \in \mathcal{J}_1} z_j C_j \sqrt{1/m}$. Putting together the above results yields

$$\texttt{CR}(\texttt{LinRP}) \leq \frac{\sum_{j \in \mathcal{J}_1} V_j + V_{\texttt{LinRP}}}{V_{\texttt{LinRP}}} \leq \frac{\sum_{j \in \mathcal{J}_1} z_j \sqrt{\frac{2CC_j}{m}}}{\sum_{j \in \mathcal{J}_1} z_j C_j \sqrt{\frac{1}{m}}} + 1 \leq \max_{j \in \mathcal{J}_1} \frac{z_j \sqrt{\frac{2CC_j}{m}}}{z_j C_j \sqrt{\frac{1}{m}}} + 1 \leq \sqrt{2\alpha} + 1. \tag{11}$$

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

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

$$\begin{aligned} V_{\texttt{LinRP}} = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_{i,j} &\geq \sum_{i=1}^{n} x_i \max_j z_{i-1,j} \sqrt{\frac{2C}{mC_j}} w_{i,j} \\ &\geq \sum_{i=1}^{n} x_i z_{i-1,j'} \sqrt{\frac{2C}{mC_{j'}}} w_{i,j'} \\ &= \sum_{i=1}^{n} z_{i-1,j'} \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2C}{mC_{j'}}} \\ &= \sum_{i=1}^{n} \left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \sqrt{m} \right\rfloor \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2C}{mC_{j'}}} \\ &\geq \sum_{l=1}^{\lfloor \sqrt{m} \rfloor - 1} l \left(\frac{C_{j'}}{\sqrt{m}} - 2\varepsilon C_{j'}\right) \sqrt{\frac{2C}{mC_{j'}}} \end{aligned} \tag{12}$$

The last inequality in the above equation uses the feature of the step function and the fact that $u_{i,j'} - u_{i-1,j'} \leq \varepsilon C_{j'}$ (see subsection A.2 in the appendix). Then, we can further lower bound the above equation as follows.

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

We have

$$\begin{aligned} \texttt{CR(LinRP)} \leq & C \left(\left(\frac{1}{\sqrt{m}} - 2\varepsilon\right) \frac{\lfloor \sqrt{m} \rfloor \left(\lfloor \sqrt{m} \rfloor - 1\right)}{2\sqrt{m}} \sqrt{2CC_{j'}} \right)^{-1} \\ \leq & \sqrt{2\alpha} \left(\left(\frac{1}{\sqrt{m}} - 2\varepsilon\right) \frac{\lfloor \sqrt{m} \rfloor \left(\lfloor \sqrt{m} \rfloor - 1\right)}{\sqrt{m}} \right)^{-1}. \end{aligned} \tag{13}$$

Comparing the results in Equations (11) and (13) completes the proof.

A proof of Theorem 4. We reuse $\mathbf{Z} = [z_1, z_2, \ldots, z_m]$ to denote the final state of the knapsack by running the ExpRP algorithm. Let $\mathcal{J}_l$, $l = 0, 1, 2, \ldots, \lfloor \log \alpha \rfloor$, be the set of dimensions satisfying $z_j \geq l$. 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 single-dimension weight of items in dimension j is less than or equal to εC_j, we have that a job is always admitted when Equation (7) holds.

First, we provide a lower bound for $V_{\texttt{ExpRP}}$ as the cumulative values of admitted items by ExpRP. By Equation (7) in ExpRP, for each admitted item, we have

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

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

$$
\begin{aligned}
V_{\mathsf{ExpRP}} = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_{i,j} &\geq \sum_{i=1}^{n}\sum_{j=1}^{m} x_i \left(2^{z_{i-1,j}} - 1\right) w_{i,j} \\
&\geq \sum_{i=1}^{n}\sum_{j=1}^{m} \left(2^{\left\lfloor \frac{u_{i-1,j}}{C_j} \log \frac{C}{C_j} \right\rfloor} - 1\right) \left(u_{i,j} - u_{i-1,j}\right) \\
&\geq \sum_{j\in\mathcal{J}_1} \sum_{l=1}^{z_j - 1} \left(C_j \log^{-1} \frac{C}{C_j} - \varepsilon C_j\right) \left(2^l - 1\right) \\
&= \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1} \frac{C}{C_j} - \varepsilon C_j\right) \left(2\left(2^{z_j - 1} - 1\right) - z_j + 1\right) \\
&= \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1} \frac{C}{C_j} - \varepsilon C_j\right) \left(2^{z_j} - z_j - 1\right).
\end{aligned} \tag{14}
$$

In addition, we have

$$
V_{\mathsf{ExpRP}} \geq \sum_{j\in\mathcal{J}_1} C_j \log^{-1} \frac{C}{C_j}.
$$

Combining the above two equations yields

$$
\begin{aligned}
V_{\mathsf{ExpRP}} \geq& \max\left\{ \sum_{j\in\mathcal{J}_1} C_j \log^{-1} \frac{C}{C_j}, \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1} \frac{C}{C_j} - \varepsilon C_j\right) \left(2^{z_j} - z_j - 1\right) \right\} \\
\geq& \beta \sum_{j\in\mathcal{J}_1} C_j \log^{-1} \frac{C}{C_j} + (1-\beta) \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1} \frac{C}{C_j} - \varepsilon C_j\right) \left(2^{z_j} - z_j - 1\right), \text{ (for any } \beta \in [0,1]).
\end{aligned} \tag{15}
$$

Next, we prove an upper bound for the total values of admitted items by any algorithm. We call an item i a Type-I item if the following holds

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

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

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

Next, we will provide an upper bound for the aggregate value of admitted Type-I items by any algorithm providing a feasible solution to OMdKP-UD.

Let i_k, $k = \{1, 2, \ldots, n_1\}$, be the indices of Type-I items. Then, by the definition of Type-I items, for any i_k, $k = \{1, 2, \ldots, n_1\}$, we have

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

It follows from the above equation that

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

where the last inequality uses the fact that $\sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \le C_j$, $j \in [m]$. By the above equation, we upper bound the total amount of values of Type-I items that are admitted by any algorithm. On the other hand, the aggregate value of Type-II items is not larger than the that of admitted items by ExpRP. Thus, the aggregate value of items admitted by any algorithm is then upper bounded by

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

Combining the results in Equation (15) with $\beta = 1/4$, the competitive ratio of ExpRP satisfies

$$\begin{aligned}
\mathsf{CR}(\mathsf{ExpRP}) \le& \frac{\sum_{j=1}^{m} \left(2^{z_j}-1\right) C_j + V_{\mathsf{ExpRP}}}{V_{\mathsf{ExpRP}}} \\
=& \frac{\sum_{j=1}^{m} \left(2^{z_j}-1\right) C_j}{V_{\mathsf{ExpRP}}} + 1 \\
\le& \frac{\sum_{j\in\mathcal{J}_1} C_j + \sum_{j\in\mathcal{J}_2} \left(2^{z_j}-1\right) C_j}{\frac{1}{4}\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\frac{C}{C_j} + \frac{3}{4}\sum_{j\in\mathcal{J}_1}\left(C_j \log^{-1}\frac{C}{C_j} - \varepsilon C_j\right)\left(2^{z_j}-z_j-1\right)} + 1 \\
\le& \max\left\{4\frac{\sum_{j\in\mathcal{J}_1} C_j}{\sum_{j\in\mathcal{J}_1} C_j \log^{-1}\frac{C}{C_j}}, \frac{4\sum_{j\in\mathcal{J}_2}\left(2^{z_j}-1\right) C_j}{3\sum_{j\in[m]}\left(C_j \log^{-1}\frac{C}{C_j} - \varepsilon C_j\right)\left(2^{z_j}-z_j-1\right)}\right\} + 1 \\
\le& \max\left\{4\log\frac{C}{C_1}, 4\frac{\log\frac{C}{C_1}}{1-\varepsilon\log\frac{C}{C_1}}\right\} + 1 = \max\left\{4\log\alpha, 4\frac{\log\alpha}{1-\varepsilon\log\alpha}\right\} + 1,
\end{aligned}$$

where the last inequality uses the fact that

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

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

Without loss of generality, we assume there exists dimension j', such that $C_{j'} - u_{n,j'} \ge \varepsilon C_{j'}$. That is $z_{j'} \ge \left\lfloor \log \frac{C}{C_{j'}} \right\rfloor - 1$.

Note that

$$u_{n,j'} \ge (1-\varepsilon) C_{j'} \ge \frac{2}{3} C_{j'}.$$

When $C/C_{j'} \leq 16$, there is

$$\texttt{CR}(\texttt{ExpRP}) \leq \frac{C}{\frac{2}{3}C_{j'}} \leq 12,$$

completing the proof. In the following, we simply assume $C/C_{j'} \geq 8$.

By the rules of the ExpRP algorithm, we have

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

where the fifth inequality uses the fact that $2^{\lfloor \log \frac{C}{C_{j'}} \rfloor} \geq 2(\lfloor \log \frac{C}{C_{j'}} \rfloor + 1)$, when $\left\lfloor \log C/C_{j'} \right\rfloor \geq 3$. Thus,

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

Concluding the above two cases yields

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

4 ONLINE ALGORITHMS FOR GENERAL OMDKP

In this section, we tackle the general OMdKP problem with arbitrary item values. Recall that in Equation (2), we defined $p_i = v_i/w_i$ as the unit value of item i. In OMdKP-UD, we assumed $p_i = 1$ and developed algorithms in Section 3. In this section, we relax this assumption and consider that p_i can take any value in $[1, \theta]$; hence, by assuming the minimum item value to be 1, parameter θ represents the ration between the maximum and minimum item values. Assuming an upper bound for item values is a common assumption for the knapsack algorithms [10, 12, 33, 39, 43], and the competitive ratio of online algorithms is characterized as a function of θ.

In this section we show that with this relaxation of arbitrary, but, bounded item values, we can extend our algorithms (both linear and exponential reservation policies) to achieve bounded competitive ratios. As the main contributions of this paper, we propose `ExpRP-G`, which extends `ExpRP`, and show it achieves the optimal competitive ratio for `OMdKP`. We proceed to present the algorithms, followed by competitive results in Section 4.2.

4.1 Online Algorithms

Similar to our proposed algorithms for `OMdKP-UD`, we investigate two types of reservation policies which involve a linear and exponential scarcity factor in their admission criteria. The proposed algorithms are called `LinRP-G` and `ExpRP-G` algorithms. Compared with `LinRP` or `ExpRP`, the new versions of the algorithms use parameter θ to scale the system states, while both use the same criteria to admit forthcoming items as `LinRP` or `ExpRP`, respectively. The `LinRP-G` and `ExpRP-G` algorithms are summarized as Algorithms 3 and 4, respectively. One can find that both `LinRP-G` and `ExpRP-G` can be reduced to `LinRP` and `ExpRP` with $\theta = 1$.

Algorithm 3 The `LinRP-G` Algorithm, upon arrival of item i

1: **Discretization:** Discretize the system state by defining $z_{i,j}$, $j \in [m]$ as

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

2: **Decision Making:** Admits item i if there is enough space to admit, i.e., $w_{i,j} \le C_j - u_{i-1,j}, \forall j \in [m]$, and the following inequality hold

$$v_i \ge \max_{j \in [m]} z_{i-1,j} \sqrt{\frac{2C}{mC_j}} w_{i,j}. \tag{16}$$

Algorithm 4 The `ExpRP-G` Algorithm, upon arrival of item i

1: **Discretization:** Discretize the system state by defining $z_{i,j}$, $j \in [m]$ as

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

2: **Decision Making:** Admits item i if there is enough space to admit, i.e., $w_{i,j} \le C_j - u_{i-1,j}, \forall j \in [m]$, and the following inequality hold

$$v_i \ge \sum_{j=1}^{m} \left(2^{z_{i-1,j}} - 1\right) w_{i,j}. \tag{17}$$

4.2 Competitive Analysis for `OMdKP`

In this section, we present a lower bound for any competitive algorithm providing a feasible solution for `OMdKP` (in Theorem 5) followed by the competitive results for `LinRP-G` (in Theorem 6) and `ExpRP-G` (in Theorem 7).

4.2.1 *Main Results.*

Theorem 5. *(Lower Bound on Competitive Ratio for* `OMdKP`*) There is no online algorithm with a competitive ratio smaller than* $\Omega(\log \theta\alpha)$ *for* `OMdKP`*.*

Theorem 6. *With $\varepsilon < 1/(2\sqrt{m})$ and $m \geq 4$, the competitive ratio of* LinRP-G *satisfies*

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

Remark. With $\varepsilon \longrightarrow 0$ and $\theta m \gg 1$, the competitive ratio of LinRP-G is $O\left(\sqrt{2\theta\alpha}\right)$.

Theorem 7. *With $\varepsilon < \min\left\{1/3, \frac{1}{2}\log^{-1}(\theta\alpha)\right\}$, the competitive ratio of* ExpRP-G *satisfies*

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

Remark. With $\varepsilon \longrightarrow 0$, the ExpRP-G algorithm attains a competitive ratio of $O(\log(\theta\alpha))$. Comparing the result in theorems 5 and 7 shows that ExpRP-G achieves the optimal competitive ratio up to a constant factor. In addition, in the special case with $m = 1$, the OMdKP problem is reduced to the basic version of online knapsack problem [12, 43], or equivalently the so-called one-way trading [15] (see [11] for the equivalence). Correspondingly, the ExpRP-G algorithm is reduced to the optimal algorithm for those two problems which use exponential thresholds to admit items.

4.2.2 Proofs. We first provide an informal proof for Theorem 5. Then, we prove the result in Theorem 7. For a the proof of Theorem 6, one can refer to Appendix B.1.

An Informal Proof for Theorem 5. We first provide an informal, yet, intuitive proof sketch for Theorem 5. Note that the OMdKP problem can be seen as an extension of either the OMdKP-UD problem (with $\theta = 1$) or the one-way trading [15] (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 OMdKP-UD problem) 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

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

A proof of Theorem 7. The proof of Theorem 7 follows similar logic to that of Theorem 4. Again, we define $\mathbf{Z} = [z_1, z_2, \ldots, z_m]$ as the ending state of the system executing the proposed ExpRP-G algorithm. Let $\mathcal{J}_l$, $l = 0, 1, 2, \ldots, \lfloor\log(\theta\alpha)\rfloor$, 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 (17) holds.

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

$$\begin{aligned}
V_{\mathsf{ExpRP\text{-}G}} &= \sum_{i=1}^{n} x_i v_i = \sum_{i=1}^{n} x_i p_i \sum_{j\in\mathcal{J}} w_{i,j} \\
&\geq \sum_{i=1}^{n} \sum_{j\in[m]} \left(2^{z_{i-1,j}} - 1\right) x_i w_{i,j} \\
&\geq \sum_{i=1}^{n} \sum_{j\in[m]} \left(2^{\left\lfloor \frac{u_{i-1,j}}{C_j} \log\left(\theta \frac{C}{C_j}\right)\right\rfloor} - 1\right) (u_{i,j} - u_{i-1,j}) \\
&\geq \sum_{j\in\mathcal{J}_1} \sum_{l=1}^{z_j-1} \left(C_j \log^{-1}\left(\theta\frac{C}{C_j}\right) - \varepsilon C_j\right)\left(2^l - 1\right) \\
&= \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1}\left(\theta\frac{C}{C_j}\right) - \varepsilon C_j\right)\left(2\left(2^{z_j-1} - 1\right) - z_j + 1\right) \\
&= \sum_{j\in\mathcal{J}_1} \left(C_j \log^{-1}\left(\theta\frac{C}{C_j}\right) - \varepsilon C_j\right)\left(2^{z_j} - z_j - 1\right).
\end{aligned} \tag{18}$$

In addition, we have another lower bound for $V_{\mathsf{ExpRP\text{-}G}}$:

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

Combining the above two equations yields

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

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

Next, we prove an upper bound for the 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.

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

Otherwise, an item is called the Type-II item. Obviously, an item is either a Type-I item or a Type-II item. In 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

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

It follows from the above equation that

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

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

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

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

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

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

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

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

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

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 $z_{j'} \geq \left\lfloor \log\left(\theta \frac{C}{C_{j'}}\right)\right\rfloor - 1$.

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

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

completing the proof.

By the rules of the ExpRP-G algorithm, we have

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

where the last inequality uses the fact that

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

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

Thus,

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

Algorithm 5 The `ExpRP-F` algorithm for fractional packing of items with arbitrary sizes

At each round i, the `ExpRP-F` algorithm determines x_i as follows:
Initialization: $x_i \leftarrow 0$, $u_j \leftarrow u_{i-1,j}$, $j \in [m]$
while $x_i < 1$ **do**
  $y \leftarrow \min_{j\in[m]} \frac{\left(\left\lfloor \frac{u_j}{C_j}\log\left(\theta\frac{C}{C_j}\right)\right\rfloor+1\right)C_j\log^{-1}\left(\theta\frac{C}{C_j}\right)-u_j}{w_{i,j}}$
  if $yv_i \geq \sum_{j=1}^{m}\left(2^{\left\lfloor \frac{u_j}{C_j}\log\left(\theta\frac{C}{C_j}\right)\right\rfloor}-1\right)yw_{i,j}$ **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

Concluding the above two cases yields

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

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` with arbitrary item sizes. Recall that the previous algorithms are designed by assuming bounded item sizes as characterized in Equation (3). For the fractional algorithm design, we relax those assumptions in this section.

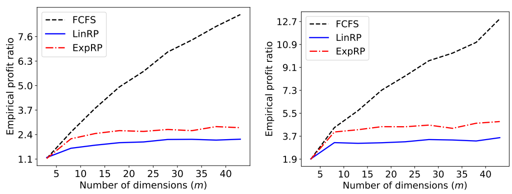
We consider the fractional model with arbitrary weights, where each item can be partially packed to a multidimensional knapsack. For brevity, we omit extending the linear reservation policy and only investigate the modified exponential reservation policy for the fractional model, called as `ExpRP-F`. `ExpRP-F` determines the admission amount of an item in an iterative manner. Specifically, it split each incoming item into multiple fractions indicated by parameter y in `ExpRP-F` and check the exponential admission criterion as used in `ExpRP-G` to admit those fractions one-by-one. The 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 5.

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

Comparing the results in theorems 4 and 7 show that `ExpRP-F` achieves a better competitive ratio than `ExpRP-G`. 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 8 is given in Appendix C.

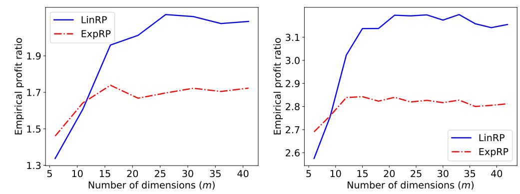
6 NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments to evaluate the proposed algorithms. Our goal is two-fold: (1) to compare the performance of both linear and exponential policies for both `OMdKP-UD` and `OMdKP` with that of `FCFS`. The results are reported in Figure 1; and (2) to compare the empirical profit ratios of linear and exponential policies under different settings with varying number of dimensions. The results are reported in Figure 2.



(a) OMdKP-UD (b) OMdKP

Fig. 1. Comparison of FCFS with our proposed algorithms for both OMdKP and OMdKP-UD



(a) OMdKP-UD (b) OMdKP

Fig. 2. Comparison between linear and exponential policies for both OMdKP and OMdKP-UD

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 theoretical competitive ratio. In all experiments, we report the average profit ratios of different algorithms for 200 random trails. For experiments in Section 6.2, the capacities over dimensions are selected between 1 to 10, and in Section 6.3, all dimension capacities are 1. For experiment in OMdKP, we set the unit-value fluctuation ratio to be $\theta = 2$, i.e., the unit values of items are randomly drawn from $[1, 2]$.

6.2 Comparison between FCFS and Proposed Online Algorithms

In this experiment, we compare the performance of linear (LinRP and LinRP-G) and exponential algorithms (ExpRP and ExpRP-G) with the FCFS strategy. We consider the following scenario for item arrivals. 100 items arrive in two batches of size 50 items. The first batch consists of items demanding a single-dimension, and items in the second batch are $m/2$-dimensional. We select weights of items uniformly at random from [0,1]. In Figure 1, we vary the number of dimensions from 3 to 45 and report the average empirical profit ratio of 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 in Figure 1(a) for OMdKP-UD and LinRP-G and ExpRP-G in Figure 1(b) for OMdKP increase slightly. The empirical profit ratio of LinRP (resp. LinRP-G) is smaller than that of ExpRP (resp. ExpRP-G), which is counter-intuitive as the worst-case theoretical analysis says otherwise. In the following, we further investigate the performance of linear and exponential algorithms using another experimental setup.

6.3 Comparison between Linear and Exponential Policies

In the second scenario, we consider 200 items partitioned into two batches. The first batch with 80 items includes only single-dimensional items while the second batch with 120 items, each item has an arbitrary dimension picked uniformly from $[3, m/2]$. Also, the item weights have been selected uniformly random. The average empirical profit ratios of linear and exponential algorithms for both models are demonstrated in Figure 2. The results shows that when m is small, e.g., $m \leq 10$, LinRP (resp. LinRP-G) outperforms ExpRP (resp. ExpRP-G) for OMdKP-UD (resp. OMdKP). However, as m becomes large, we see the exponential algorithms outperform the linear ones for both models. Comparing the constant values in theoretical competitive ratios, we see the same behavior of numerical and theoretical results.

7 RELATED WORK

The offline version of OMdKP is a well-studied problem in different settings in literature [16, 17, 23, 31, 36, 38]. Our problem is similar to 0-1 version [8, 17] 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 [20], resource allocation [19]. Nevertheless, to the best of our knowledge, the problem has not studies in the online setting. We note that recently in literature some generic online resource allocation problems have been studied [4, 5], that our problem could be expressed as their special version. However, our competitive ratios are optimal in this paper, while applying the results in [4, 5] can not lead to optimal competitive ratios for our problem.

Another category of similar problems is the online multiple knapsack problem (OMKP) [10, 12, 22, 25, 29, 33, 43]. 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 [18, 26, 30], cloud computing jobs [24, 27, 42], admission control for electric vehicles at charging stations [2, 33, 35], and QoS buffer management [37]. 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.

8 CONCLUSION

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 matches the lower bounds up to a constant factor. 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.

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, 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.

[5] 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).

[6] Marcin Bienkowski, Maciej Pacut, and Krzysztof Piecuch. 2020. An Optimal Algorithm for Online Multiple Knapsack. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

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

[8] 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

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

[10] 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

[11] 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.

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

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

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

[15] 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.

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

[17] 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

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

[19] 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.

[20] 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.

[21] Xin Han, Yasushi Kawase, and Kazuhisa Makino. 2015. Randomized algorithms for online knapsack problems. *Theoretical Computer Science* 562 (2015), 395–405.

[22] 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

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] Will Ma, David Simchi-Levi, and Jinglong Zhao. 2019. The Competitive Ratio of Threshold Policies for Online Unit-density Knapsack Problems. *Available at SSRN 3423199* (2019).

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

[30] 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.

[31] 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.
[32] 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.
[33] 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.
[34] 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.
[35] 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.
[36] M Jalali Varnamkhasti. 2012. Overview of the algorithms for solving the multidimensional knapsack problems. *Advanced Studies in Biology* 4, 1 (2012), 37–47.
[37] 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.
[38] Quan Yuan and Zhixin Yang. 2013. An Improved Weight-coded Evolutionary Algorithm for the Multidimensional Knapsack Problem. *arXiv preprint arXiv:1302.5374* (2013).
[39] Ali Zeynali, Bo Sun, Mohammad Hajiesmaili, and Adam Wierman. 2021. Data-driven Competitive Algorithms for Online Knapsack and Set Cover. In *Proc. of AAAI*.
[40] 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.
[41] 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.
[42] 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.
[43] 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 PROOFS FOR OMDKP-UD

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 the following weights to the FCFS algorithm for n times.

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

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

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

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 C. In this way, we show that the competitive ratio of the FCFS strategy is at least $C/C_1 = \alpha$, completing the proof.

A.2 Proof of the Last Inequality in Equation (12)

We use Figure 3 to facilitate our proof. Specifically, $\sum_{i=1}^{n} \left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \sqrt{m} \right\rfloor \left(u_{i,j'} - u_{i-1,j'}\right)$ can be seen as an approximation of the integral of the step function $\left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \sqrt{m} \right\rfloor$ with step length being $\frac{C_{j'}}{\sqrt{m}}$, and be visualized by the colored area in Figure 3. By calculating a lower bound for the size of the colored area, we prove the last inequality in Equation (12). We note that the methodology used above is also applied to other proofs in the paper, e.g., those for Equation (14) and (18) etc.

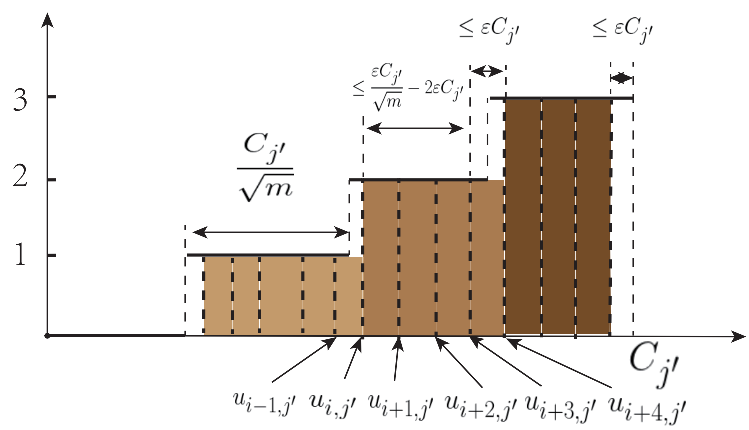


Fig. 3. Visualized proof for Equation (12)

A.3 Additional Lemmas in Section 3

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

$$\sum_{l'=1}^{l} l' y_{l'} > \frac{lC_1}{\delta},$$

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 $A_l := \sum_{l'=1}^{l} l' y_{l'}$. 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 PROOFS FOR SECTION 4

B.1 The Proof of Theorem 6

A proof of Theorem 6. Let $\mathbf{z} = [z_1, z_2, \dots, z_m]$ be the final state of the system executing the `LinRP-G` algorithm, where $z_j = z_{n,j}, j \in [m]$. Let $\mathcal{J}_l$, $l = \{0, 1, 2, \dots, \lfloor \sqrt{\theta m} \rfloor\}$, 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 (16) is satisfied. Consider the following constraints for incoming item i

$$
\text{constraint} - j : v_i \geq z_j \sqrt{\frac{2C}{mC_j}} w_{i,j}, \; j \in \mathcal{J}_1. \tag{20}
$$

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, $k = \{1, 2, \dots, n_j\}$, be the k-th admitted Type-j item by any algorithm. By the definition and using Equation (20), we upper bound the aggregate value of admitted Type-j items, denoted as V_j, as follows

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

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

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 (20), it will be admitted by `LinRP-G`. Thus, the aggregate value of the admitted items satisfying all the constraints in Equation (20), is not greater than $V_{\texttt{LinRP-G}}$, as the aggregate value obtained by `LinRP-G`. 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 $\sum_{j \in \mathcal{J}_1} V_j + V_{\texttt{LinRP-G}}$. We also have

$V_{\texttt{LinRP}} \geq \sum_{j\in\mathcal{J}_1} z_j C_j \sqrt{1/(\theta m)}$. Putting together the above results yields

$$\texttt{CR}(\texttt{LinRP-G}) \leq \frac{\sum_{j\in\mathcal{J}_1} V_j + V_{\texttt{LinRP-G}}}{V_{\texttt{LinRP-G}}} \leq \frac{\sum_{j\in\mathcal{J}_1} z_j \sqrt{\frac{2CC_j}{m}}}{\sum_{j\in\mathcal{J}_1} z_j C_j \sqrt{\frac{1}{\theta m}}} + 1 \leq \max_{j\in\mathcal{J}_1} \frac{z_j \sqrt{\frac{2CC_j}{m}}}{z_j C_j \sqrt{\frac{1}{\theta m}}} + 1 \leq \sqrt{2\theta\alpha} + 1. \tag{21}$$

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 $z_{j'} \geq \left\lfloor \sqrt{\theta m} \right\rfloor - 1$. The aggregate value of items admitted by LinRP-G is

$$\begin{aligned} V_{\texttt{LinRP-G}} = \sum_{i=1}^{n} x_i v_i &\geq \sum_{i=1}^{n} x_i \max_j z_{i-1,j} \sqrt{\frac{2C}{mC_j}} w_{i,j} \\ &\geq \sum_{i=1}^{n} x_i z_{i-1,j'} \sqrt{\frac{2C}{mC_{j'}}} w_{i,j'} \\ &= \sum_{i=1}^{n} z_{i-1,j'} \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2C}{mC_{j'}}} \\ &= \sum_{i=1}^{n} \left\lfloor \frac{u_{i-1,j'}}{C_{j'}} \sqrt{\theta m} \right\rfloor \left(u_{i,j'} - u_{i-1,j'}\right) \sqrt{\frac{2C}{mC_{j'}}} \\ &\geq \sum_{l=1}^{\left\lfloor \sqrt{\theta m} \right\rfloor - 1} l \left(\frac{C_{j'}}{\sqrt{\theta m}} - 2\varepsilon C_{j'} \right) \sqrt{\frac{2C}{mC_{j'}}} \end{aligned}$$

The last inequality in the above equation uses the feature of the step function and the fact that $u_{i,j'} - u_{i-1,j'} \leq \varepsilon C_{j'}$. Then, we can further lower bound the above equation as follows.

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

We have

$$\begin{aligned} \texttt{CR}(\texttt{LinRP-G}) &\leq \theta C \left(\left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon \right) \frac{\left\lfloor \sqrt{\theta m} \right\rfloor \left(\left\lfloor \sqrt{\theta m} \right\rfloor - 1 \right)}{2\sqrt{m}} \sqrt{2CC_{j'}} \right)^{-1} \\ &\leq \sqrt{2\alpha} \left(\left(\frac{1}{\sqrt{\theta m}} - 2\varepsilon \right) \frac{\left\lfloor \sqrt{\theta m} \right\rfloor \left(\left\lfloor \sqrt{\theta m} \right\rfloor - 1 \right)}{\theta\sqrt{m}} \right)^{-1}. \end{aligned} \tag{22}$$

Comparing the results in Equations (21) and (22) completes the proof.

C PROOF FOR THE FRACTIONAL RESULT IN THEOREM 8

Let $\mathbf{Z} = [z_1, z_2, \ldots, z_m]$ be the ending state of the system executing the proposed ExpRP-F algorithm. Let $\mathcal{J}_l$, $l = 0, 1, 2, \ldots, \lfloor \log(\theta\alpha) \rfloor$, 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 $y_{i,r}$, $r = 1, 2, \ldots, n_i$ 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 $u_{i-1,j} + \sum_{k=1}^{r} y_{i,k} w_{i,j}$, $r = 1, 2, \ldots, n_i$.

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 $V_{\texttt{ExpRP-F}}$.

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

In addition, we have

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

Combining the above two equations yields

$$
\begin{aligned}
V_{\texttt{ExpRP-F}} \geq& \max\left\{ \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta \frac{C}{C_j}\right), \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta \frac{C}{C_j}\right) \left(2^{z_j} - z_j - 1\right) \right\} \\
\geq& \beta \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta \frac{C}{C_j}\right) + (1-\beta) \sum_{j\in\mathcal{J}_1} C_j \log^{-1}\left(\theta \frac{C}{C_j}\right) \left(2^{z_j} - z_j - 1\right),
\end{aligned}
\tag{23}
$$

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

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

where the last inequality uses the fact that $\sum_{k=1}^{n_1} x_{i_k} w_{i_k,j} \le C_j$, $j = 1, 2, \ldots, m$. 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

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

Combined with the results in Equation (23) with $\beta = 1/4$, the competitive ratio of the ExpRP-F algorithm satisfies

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

where the last inequality uses the fact that

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

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 $\theta \frac{C}{C_{j'}} \geq 8$. Otherwise, there is

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

By the rules of the ExpRP algorithm, we have

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

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

Thus,

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

Concluding the above two cases yields

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