Online Multidimensional Packing Problems in the Random-Order Model

David Naori
Computer Science Department, Technion, 32000 Haifa, Israel
dnaori@cs.technion.ac.il

Danny Raz
Computer Science Department, Technion, 32000 Haifa, Israel
danny@cs.technion.ac.il

Abstract

We study online multidimensional variants of the generalized assignment problem which are used to model prominent real-world applications, such as the assignment of virtual machines with multiple resource requirements to physical infrastructure in cloud computing. These problems can be seen as an extension of the well known secretary problem and thus the standard online worst-case model cannot provide any performance guarantee. The prevailing model in this case is the random-order model, which provides a useful realistic and robust alternative. Using this model, we study the d-dimensional generalized assignment problem, where we introduce a novel technique that achieves an $O(d)$-competitive algorithms and prove a matching lower bound of $\Omega(d)$. Furthermore, our algorithm improves upon the best-known competitive-ratio for the online (one-dimensional) generalized assignment problem and the online knapsack problem.

2012 ACM Subject Classification Theory of computation → Packing and covering problems; Theory of computation → Online algorithms

Keywords and phrases Random Order, Generalized Assignment Problem, Knapsack Problem, Multidimensional Packing, Secretary Problem

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2019.10

1 Introduction

Online multidimensional packing problems appear in a wide verity of real-world applications [8]. A recent relevant example is the assignment of virtual elements to the physical infrastructure in Network Function Virtualization (NFV) and cloud computing (see [20, 21] for example). Typically, in these problems, we are given a set of bins, each with a certain capacity profile, then, items arrive one-by-one in an online fashion, each with a certain size and profit. Upon each arrival, one has to decide immediately and irrevocably whether and where to pack the current item. The goal is to find an assignment that maximizes the total profit without exceeding the capacity of any bin. These problems can be viewed as generalizations of the well-known secretary problem, in which we have a single bin, and every secretary consumes the capacity of the whole bin (see [6] for a formal definition of the secretary problem).

The common way of analyzing online algorithms is to use the worst-case model, where an adversary picks an instance along with the order in which items are revealed to the online algorithm. Despite its prevalence in the analysis of online algorithms, this setting is too pessimistic for the problem at hand. Indeed, no online algorithm can achieve any non-trivial worst-case competitive-ratio, even for the simple case of the secretary problem, as shown by Aggarwal et al. [1]. A more realistic model is the random-order model in which the power of choosing the arrival order of items is taken away from the adversary, instead, the

arrival order is chosen uniformly at random. In this model, we say that an algorithm ALG is c-competitive if for every input instance $\mathcal{I}$ it holds that, $c \cdot \mathrm{E}[\mathrm{ALG}(\mathcal{I})] \geq \mathrm{OPT}(\mathcal{I})$, where the expectation is taken over the random arrival orders and the randomness of the algorithm.[1]

Kesselheim et al. [13, 14] generalized the known optimal algorithm for the secretary problem to various packing problems in the random-order model. The outline of the generic algorithmic framework is as follows: it starts with a sampling phase in which the algorithm only observes the arriving items. Then, at every subsequent online round, the algorithm computes a local solution for the sub-instance consisting of all the items that arrived so far. If the bin in which the current item is packed in this local solution has enough free capacity (i.e., an assignment of the current item in this bin is feasible) the algorithm carries it out, otherwise, it leaves the item unpacked. Using this framework, Kesselheim et al. [14] presented an algorithm for the online generalized assignment problem (GAP) with the best-known competitive-ratio (prior to this work).

In GAP we have a set of bins and a set of items. Each bin has a certain non-negative capacity and each item has several packing options, one for each bin. Each packing option is associated with a certain consumption from the capacity of the bin and a certain profit it provides. The goal is to pack the items in the bins where each item can be packed at most once, maximizing the total profit without exceeding the capacity of any bin. A major challenge in online packing problems, and online GAP in particular, is to handle both items with high consumption of resources compared to a bin capacity, as well as items with low consumption. Kesselheim et al. handle this challenge by partitioning a GAP instance into two sub-instances: the first contains all "heavy" packing options of items, that is, packing options that occupy more than half of a bin capacity, the second is the complementary sub-instance that contains all "light" packing options. Their algorithm makes an initial random choice to operate on one of the sub-instances exclusively. Although it achieves the best-known results, this behaviour is undesirable for most applications, since it always leaves one type of items unpacked.

We use a similar algorithmic framework to design an online algorithm for the d-dimensional generalization of online GAP, or online Vector Generalized Assignment Problem (VGAP), in which the capacity profile of each bin, as well as the consumption of items from each bin, is described by a d-dimensional vector. The goal remains to maximize the profit, while the capacity of each bin must not be exceeded in any of its d dimensions. To the best of our knowledge, this is the first time the online version of this problem is studied. Our algorithm offers a preferable behaviour and improves upon the best-known competitive-ratio for online GAP. To achieve this, we take a different approach to overcome the challenge: instead of limiting the algorithm to either "heavy" or "light" packing options, our algorithm considers them both. It operates in three phases: a sampling phase, a phase for "heavy" packing options, and a phase for "light" packing options. To compute the tentative assignments, our algorithm in the second phase uses maximum-weight bipartite matching, and in the third phase, it uses an optimal fractional solution for the LP-relaxation of the local problem, and randomized rounding.

We also apply our technique to the $\{0,1\}$-VGAP in which every packing option of an item in every dimension must consume either the whole capacity of the bin or non of it. In one-dimension this problem is identical to weighted bipartite matching. For $\{0,1\}$-VGAP we partition the instance by a different criterion: the number of non-zero entries in the consumption vector of a packing option.

[1] We follow the definition used in [4, 5, 13] although it is also common to refer to such algorithm as $1/c$-competitive.

Another interesting special case of VGAP is the Vector Multiple Knapsack Problem (VMKP), in which all bins are identical, and the packing options of each item are identical for all bins. That is, regardless of the bin's identity, the item consumes the same amount of capacity and raises the same profit. For instances of VMKP with at least two bins, we describe a simpler algorithm that avoids partitioning the instance. Here, our algorithm uses a fractional solution for the LP-relaxation of the local problem only to make a binary decision whether to pack the current item or not. For the actual packing, it exploits the fact that all packing options are identical and uses greedy First Fit approach, typically used for the Bin Packing problem.

Finally, we prove a lower bound for the online vector knapsack problem in the random-order model, which also applies to VMKP and VGAP, and indicates that our algorithms are asymptotically optimal. This lower bound is inspired by the work of Babaioff et al. [5] on the matroid secretary problem, which is based solely on the inherent uncertainty due to the online nature of the problem without any complexity assumptions.

Our main contributions are:

1. We describe an algorithm for online VGAP with a competitive-ratio of $\sqrt[4]{e}\,(4d+2) \approx 5.14d + 2.57$, where d is the dimension. For the VMKP with at least two bins we describe a $(4d+2)$-competitive algorithm. To the best of our knowledge, these problems are studied for the first time.
2. We prove a matching lower bound of $\Omega(d)$ which is valid both for VGAP and VMKP.
3. Our method improves upon the best-known competitive-ratio for (one-dimensional) GAP from 8.1 to 6.99 (which is also the best-known competitive-ratio for online knapsack).

2 Related Work

Online packing problems in the random-order model have been studied extensively in recent years, most of them are generalizations of the secretary problem which has an optimal e-competitive algorithm [10, 17]. An immediate generalization is the multiple-choice secretary problem, in which one is allowed to pick up to k secretaries. It was studied by Kleinberg [15], where he presented an asymptotically optimal $\frac{\sqrt{k}}{\sqrt{k}-5}$-competitive algorithm. Another related problem is the weighted-matching problem which has an optimal e-competitive algorithm by Kesselheim et al. [13].

The online knapsack problem, which generalizes the multiple-secretary problem, was studied by Babaioff et al. [4] who presented an $10e$-competitive algorithm. It was later improved by the work of Kesselheim et al. [14] on online GAP, which generalizes all of the above problems. They presented an 8.1-competitive algorithm which is the best-known competitive-ratio for online GAP and the online knapsack problem. Our result for VGAP improves on that.

In their work, Kesselheim et al. also studied the online packing LPs problem with column sparsity d. The general online packing LPs problem was studied before by [2, 11, 19]. In this problem, there is a set of resources and a set of requests. Each request has several options to be served and each option is associated with a profit and a certain demand from each resource. For column sparsity d, each request may have a demand from at most d of the resources. This problem generalizes VGAP studied in this paper, however, to the best of our knowledge, the only known competitive online algorithms for this problem are for the special case of $B \geq 2$, where B is the capacity ratio, i.e., the minimal ratio between the capacity of a resource and the maximum demand for this resource. For this case they presented an $O\left(d^{1/(B-1)}\right)$-competitive algorithm which in case $B = \Omega\left(\log d/\epsilon^2\right)$ is $(1+\epsilon)$-competitive.

Dean et al. [9] showed that under the assumption of NP $\neq$ ZPP, the packing integer programs problem (PIP, also known as vector knapsack) which is a special case of VMKP, cannot be approximated in polynomial time to within $d^{1-\epsilon}$ for any $\epsilon > 0$ even in the offline settings. Under the same assumptions, Chekuri et al. [7] showed that the $\{0,1\}$-case cannot be approximated to within $d^{1/2-\epsilon}$ for any $\epsilon > 0$. Their results are also applicable for the offline VMKP, VGAP and the $\{0,1\}$-VGAP. By using the results of Zuckerman [22], the same hardness result can be proved under the weaker assumption of P $\neq$ NP instead. As opposed to these results, our lower bound holds with no complexity assumptions, and applies even for algorithms with unbounded computational power.

Some related problems have competitive algorithms in the worst-case model too. One example is the AdWords problem which is a special case of GAP in which the profit of each item is equal to its size. Under the assumption that items are small compared to the capacity of the bins, Metha et al. [18] presented an optimal $\frac{e}{e-1}$-competitive algorithm. Without this assumption, the best known competitive-ratio is 2 [16]. Another example is the online vector bin packing problem, in which items arrive one-by-one, and the goal is to pack them all in the minimum number of unit sized d-dimensional bins. This problem was studied by Garey et al. [12] who showed that the First Fit algorithm has a worst-case competitive-ratio of $(d + 0.7)$. More recently, Azar et al. [3] showed that this algorithm is asymptotically optimal by proving a lower bound of $\Omega\left(d^{1-\epsilon}\right)$.

3 Vector Generalized Assignment Problem

In the d-dimensional *Generalized Assignment Problem* (VGAP), we have a set of m d-dimensional bins and a set of n d-dimensional items that may be packed in the bins. Each bin j has a capacity $\mathbf{b}_j = \left(b_j^1, \ldots, b_j^d\right) \in \mathbb{R}^d_{\geq 0}$. Packing item i in bin j consumes an amount of $\mathbf{w}_{i,j} = \left(w_{i,j}^1, \ldots, w_{i,j}^d\right) \in \mathbb{R}^d_{\geq 0}$ from bin's j capacity and provides a profit of $p_{i,j} \geq 0$. Each item may be packed in at most one of the bins and the capacity of each bin must not be exceeded in any of its d dimensions. The goal is to find a feasible packing that maximizes the total profit. We use the following LP-formulation:

$$
\begin{aligned}
\max \quad & \sum_{i\in[n], j\in[m]} p_{i,j} x_{i,j} \\
\text{s.t.} \quad & \sum_{i\in[n]} w_{i,j}^t x_{i,j} \leq b_j^t, && j \in [m], t \in [d] \\
& \sum_{j\in[m]} x_{i,j} \leq 1, && i \in [n] \\
& x_{i,j} \in \{0,1\}, && i \in [n], j \in [m].
\end{aligned}
$$

We consider the online version of the problem in which the set of bins and their capacities are initially known, as well as the total number of items n. The items, however, arrive one by one in a random order. When item i arrives, we learn its *packing options*, i.e., its consumption on every bin $\mathbf{w}_{i,1}, \ldots, \mathbf{w}_{i,m}$ (which we also call the *weight vectors* of i) along with the corresponding profits $p_{i,1}, \ldots, p_{i,m}$. After every arrival, an immediate and irrevocable decision must be made: Assign the item to one of the available bins or leave the item unpacked.

Our algorithm is based on the technique presented by the authors of [14] with several critical improvements (see Algorithm 1). We call the packing option of item i in bin j *light* if $w_{i,j}^t \leq b_j^t/2$, $\forall t \in [d]$, otherwise, we call it *heavy*. Given a GAP instance $\mathcal{I}$ we partition it

into two sub-instances $\mathcal{I}_{heavy}$ and $\mathcal{I}_{light}$, both consist of the original items and bins, however, $\mathcal{I}_{heavy}$ consists only of the heavy packing options of every item, while $\mathcal{I}_{light}$ consists only of the light ones. In contrast to the algorithm presented in [14] that makes a random choice whether to operate on $\mathcal{I}_{heavy}$ or $\mathcal{I}_{light}$ exclusively, our algorithm considers them both. It is based on the intuition that heavy options may need a chance to be packed first, since any other packing decision might prevent them from being packed, while light options are more likely to fit in. Our algorithm operates in three phases: the *sampling phase* in which it only observes the arriving items, the *heavy phase* in which it considers only heavy options, and the *light phase* in which it considers only light options. In the heavy phase, our algorithm uses a matching in a weighted bipartite graph to make packing decisions, to this end, given an instance $\mathcal{I}$ we define a weighted bipartite graph $G(\mathcal{I}) = (L, R, E)$, where L is the set of items of $\mathcal{I}$, R is the set of bins of $\mathcal{I}$, and there exists an edge $(i, j) \in E$ of weight $p_{i,j}$ if item i can be packed in bin j (i.e., $w^t_{i,j} \leq b^t_j$, $\forall t \in [d]$). Each phase takes place in a continuous fraction of the online rounds. To partition the rounds into phases, we use two parameters q_1 and q_2 that will be defined thereafter. For convenience of presentation and analysis, we represent a packing by a set $\mathrm{P} \subseteq [n] \times [m]$ such that $\mathrm{P} = \{(i, j) : i \text{ is packed in bin } j\}$. We also define $p_{i,0} = 0$, $\forall i \in [n]$. For an instance $\mathcal{I}$ and a subset S of its items, we denote by $\mathcal{I}|_S$ the sub-instance that consists only of the items in S.

Algorithm 1 Online VGAP.

$S_0 \leftarrow \emptyset$, $\mathrm{P}_0 \leftarrow \emptyset$;
for *each item* i_ℓ *that arrives at round* ℓ **do**
  $S_\ell \leftarrow S_{\ell-1} \cup \{i_\ell\}$;
  if $\ell \leq q_1 n$ **then** `/* sampling phase */`
    continue to the next round;
  else if $q_1 n + 1 \leq \ell \leq q_2 n$ **then** `/* heavy phase */`
    Let $x^{(\ell)}$ be a maximum-weight matching in $G(\mathcal{I}_{heavy}|_{S_\ell})$;
    `// compute a tentative assignment` (i_ℓ, j_ℓ)
    if i_ℓ *is matched in* $x^{(\ell)}$ **then**
      Let j_ℓ be the bin to which i_ℓ is matched;
    else
      $j_\ell \leftarrow 0$
    if $j_\ell \neq 0$ ***and*** j_ℓ *is empty in* $\mathrm{P}_{\ell-1}$ **then**
      $\mathrm{P}_\ell \leftarrow \mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$;
  else `//` $(\ell \geq q_2 n + 1)$ `/* light phase */`
    Let $x^{(\ell)}$ be an optimal fractional solution for the LP-relaxation of $\mathcal{I}_{light}|_{S_\ell}$;
    `// compute a tentative assignment` (i_ℓ, j_ℓ) `by randomized rounding`
    Choose bin j_ℓ randomly where $\Pr[j_\ell = j] = x^{(\ell)}_{i_\ell,j}$ and
    $\Pr[j_\ell = 0] = 1 - \sum_{j \in [m]} x^{(\ell)}_{i_\ell,j}$;
    if $j_\ell \neq 0$ ***and*** $\mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$ *is feasible* **then**
      $\mathrm{P}_\ell \leftarrow \mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$;
return P_n

We now analyze the performance of Algorithm 1. Let $\mathrm{OPT}(\mathcal{I})$ and $\mathrm{ALG}(\mathcal{I})$ denote the overall profit of the optimal packing and the overall profit of the packing produced by Algorithm 1 on instance $\mathcal{I}$ respectively. Let R_ℓ denote the profit raised by the algorithm

at round ℓ. In Lemma 1 and Lemma 2 below, we bound the expected profit raised at each round of the heavy phase and the light phase respectively. Similar claims are presented in [13] and [14].

▶ **Lemma 1.** *For $q_1 n + 1 \leq \ell \leq q_2 n$, we have $\mathrm{E}[R_\ell] \geq \frac{q_1}{\ell-1} \cdot \frac{1}{d}\mathrm{OPT}(\mathcal{I}_{heavy})$.*

Proof. Let x^* be an optimal solution for $\mathcal{I}_{heavy}$, hence, $p^T x^* = \mathrm{OPT}(\mathcal{I}_{heavy})$, and let $x^*|_{S_\ell}$ denote the projection of x^* onto the set of items S_ℓ, i.e., $(x^*|_{S_\ell})_{i,j} = x^*_{i,j}$ if $i \in S_\ell$ and $(x^*|_{S_\ell})_{i,j} = 0$ otherwise. Observe (by the definition of heavy) that in $x^*|_{S_\ell}$ every bin holds at most d items. Let x^*_ℓ be the solution obtained from $x^*|_{S_\ell}$ by leaving only the most profitable item in each bin. We get $p^T x^*_\ell \geq \frac{1}{d} \cdot p^T(x^*|_{S_\ell})$. Also, since x^*_ℓ is a feasible matching in $G(\mathcal{I}_{heavy}|_{S_\ell})$, we have $p^T x^{(\ell)} \geq p^T x^*_\ell \geq \frac{1}{d} \cdot p^T(x^*|_{S_\ell})$. Now since $S_\ell \subseteq [n]$ is a uniformly random subset of size ℓ, we have $\mathrm{E}\left[p^T(x^*|_{S_\ell})\right] = \frac{\ell}{n} \cdot \mathrm{OPT}(\mathcal{I}_{heavy})$. Also, i_ℓ can be viewed as a uniformly random item of S_ℓ, and since $x^{(\ell)}$ is a matching we have $\mathrm{E}[p_{i_\ell,j_\ell}] = \mathrm{E}\left[\sum_{j\in[m]} x^{(\ell)}_{i_\ell,j} p_{i_\ell,j}\right] = \frac{1}{\ell}\mathrm{E}\left[p^T x^{(\ell)}\right]$. Combining the results together, we get

$$\mathrm{E}[p_{i_\ell,j_\ell}] = \frac{1}{\ell}\mathrm{E}\left[p^T x^{(\ell)}\right] \geq \frac{1}{\ell}\mathrm{E}\left[\frac{1}{d} \cdot p^T(x^*|_{S_\ell})\right] = \frac{1}{n \cdot d}\mathrm{OPT}(\mathcal{I}_{heavy}).$$

The above expectation is taken only over the random choice of the subset $S_\ell \subseteq [n]$ and the random choice of $i_\ell \in S_\ell$, while the arrival order of items in previous rounds is irrelevant. We now bound the probability of successful assignment over the random arrival order of previous items. The assignment is successful if no item is packed in j_ℓ in rounds $q_1 n, \ldots, \ell - 1$. At round $\ell - 1$ the algorithm uses a maximum-weight matching in $G\left(\mathcal{I}_{heavy}|_{S_{\ell-1}}\right)$ to compute a tentative assignment $(i_{\ell-1}, j_{\ell-1})$. In that matching at most one item is matched to j_ℓ. Since $i_{\ell-1}$ is a uniformly random item of $S_{\ell-1}$, the probability that $i_{\ell-1}$ is matched to j_ℓ is at most $1/(\ell-1)$ regardless of the arrival order of the items in rounds $1, \ldots, \ell-2$, hence, we can treat subsequent events as independent and repeat the argument inductively from $\ell - 1$ to $q_1 n + 1$ to get that the probability of successful assignment is at least $\prod_{k=q_1 n+1}^{\ell-1}\left(1 - \frac{1}{k}\right) = \frac{q_1 n}{\ell-1}$. By combining the expected profit with the probability of successful assignment, we get the lemma. ◀

▶ **Lemma 2.** *For $\ell \geq q_2 n + 1$, we have $\mathrm{E}[R_\ell] \geq \frac{q_1}{q_2}\left(1 - 2d\sum_{k=q_2 n+1}^{\ell-1}\frac{1}{k}\right)\frac{1}{n}\mathrm{OPT}(\mathcal{I}_{light})$.*

Proof. Let x^* be an optimal solution for $\mathcal{I}_{light}$. At round $\ell \geq q_2 n + 1$ the algorithm uses randomized rounding to determine the tentative assignment of i_ℓ from the fractional LP-solution $x^{(\ell)}$, therefore, $\mathrm{E}[p_{i_\ell,j_\ell}] = \mathrm{E}\left[\sum_{j\in[m]} x^{(\ell)}_{i_\ell,j} p_{i_\ell,j}\right]$. Using this observation, we can now follow a similar argument to that in the proof of Lemma 1 and get that for $\ell \geq q_2 n + 1$, we have $\mathrm{E}[p_{i_\ell,j_\ell}] = \frac{1}{\ell}\mathrm{E}\left[p^T x^{(\ell)}\right] \geq \frac{1}{\ell}\mathrm{E}\left[p^T(x^*|_{S_\ell})\right] = \frac{1}{n}\mathrm{OPT}(\mathcal{I}_{light})$, where the expectation is taken only over the random choice of the subset $S_\ell \subseteq [n]$, the random choice of $i_\ell \in S_\ell$ and the internal randomness of the algorithm at round ℓ. Here too, we bound the probability of successful assignment over the random arrival order of previous items and the internal randomness of the algorithm in previous rounds. Let us denote by $c(j,t,\ell)$ the total consumption of tentative assignments to bin j in dimension t during the light phase and before round ℓ. At round ℓ, the algorithm considers only light options, therefore, the assignment of i_ℓ to j_ℓ must be successful if the following conditions hold: (1) no item was packed in j_ℓ during the heavy phase, and (2) for every dimension $t \in [d]$, $c(j_\ell, t, \ell) \leq b^t_{j_\ell}/2$. Let us denote event (1) by H_ℓ, and the events described in (2) by L^t_ℓ for every dimension $t \in [d]$. We now bound $\mathrm{E}[c(j_\ell, t, \ell)]$ for every $t \in [d]$. Fix $t \in [d]$, at round $k < \ell$ of the light phase, the algorithm computes a tentative assignment based on a fractional optimal

solution for the LP-relaxation of $\mathcal{I}_{light}|_{S_k}$. In that solution, the total consumption of bin j_ℓ in dimension t is at most $b^t_{j_\ell}$. Since i_k can be viewed as a uniformly random item of S_k, the expected consumption of i_k from j_ℓ in dimension t is at most $b^t_{j_\ell}/k$, where the expectation is taken over the choice of $i_k \in S_k$ and the internal randomness of the algorithm at round k. Therefore, it is independent of the arrival order of items in rounds $1,\ldots,k-1$, and the internal randomness used in those rounds. Hence, $\mathrm{E}\left[c\left(j_\ell,t,\ell\right)\right] \leq \sum_{k=q_2n+1}^{\ell-1} b^t_{j_\ell}/k$. We have

$$\begin{aligned}\Pr\left[\bigwedge_{t=1}^{d} L^t_\ell\right] &= 1-\Pr\left[\bigvee_{t=1}^{d}\neg L^t_\ell\right] \geq 1-\sum_{t=1}^{d}\Pr\left[\neg L^t_\ell\right]\\ &\geq 1-\sum_{t=1}^{d}\frac{\sum_{k=q_2n+1}^{\ell-1} b^t_{j_\ell}/k}{b^t_{j_\ell}/2} \geq 1-2d\sum_{k=q_2n+1}^{\ell-1}\frac{1}{k}.\end{aligned}$$

The first inequality is due to a union bound, and the second is due to Markov's inequality. Since this event is independent of the arrival order of items in the heavy phase, we can follow the argument from the proof of the previous lemma and get that the probability of succesful assignment is at least

$$\Pr\left[H_\ell \wedge \bigwedge_{t=1}^{d} L^t_\ell\right] \geq \prod_{k=q_1n+1}^{q_2n}\left(1-\frac{1}{k}\right)\left(1-2d\sum_{k=q_2n+1}^{\ell-1}\frac{1}{k}\right) = \frac{q_1}{q_2}\left(1-2d\sum_{k=q_2n+1}^{\ell-1}\frac{1}{k}\right).$$

We can now combine the results of the expected profit and the success probability to get the lemma. ◀

▶ **Theorem 3.** *For $q_2 = 2d/(2d+1)$ and $q_1 = q_2/\sqrt[4]{e}$, Algorithm 1 is $\sqrt[4]{e}(4d+2)$-competitive.*

Proof. The overall profit of the algorithm can be written as $\mathrm{E}\left[\mathrm{ALG}\right] = \sum_{\ell=1}^{n}\mathrm{E}\left[R_\ell\right]$. We sum over the profit raised in each phase separately. For the heavy phase we have

$$\begin{aligned}\sum_{\ell=q_1n+1}^{q_2n}\mathrm{E}\left[R_\ell\right] &\geq \sum_{\ell=q_1n+1}^{q_2n}\frac{q_1}{\ell-1}\cdot\frac{1}{d}\mathrm{OPT}\left(\mathcal{I}_{heavy}\right)\\ &= \mathrm{OPT}\left(\mathcal{I}_{heavy}\right)\frac{q_1}{d}\sum_{\ell=q_1n}^{q_2n-1}\frac{1}{\ell} \geq \mathrm{OPT}\left(\mathcal{I}_{heavy}\right)\frac{q_1}{d}\ln\left(\frac{q_2}{q_1}\right).\end{aligned}$$

The first inequality follows from Lemma 1 and the second inequality is due to the fact that $\sum_{\ell=q_1n}^{q_2n-1}\frac{1}{\ell} \geq \int_{q_1n}^{q_2n}\frac{1}{x}dx = \ln\left(\frac{q_2}{q_1}\right)$. For the light phase we have

$$\begin{aligned}\sum_{\ell=q_2n+1}^{n}\mathrm{E}\left[R_\ell\right] &\geq \sum_{\ell=q_2n+1}^{n}\frac{q_1}{q_2}\left(1-2d\sum_{k=q_2n+1}^{\ell-1}\frac{1}{k}\right)\frac{1}{n}\mathrm{OPT}\left(\mathcal{I}_{light}\right)\\ &= \frac{1}{n}\mathrm{OPT}\left(\mathcal{I}_{light}\right)\frac{q_1}{q_2}\left(\left(1-q_2\right)n-2d\sum_{k=q_2n+1}^{n}\left(\frac{n}{k}-1\right)\right)\\ &\geq \mathrm{OPT}\left(\mathcal{I}_{light}\right)\frac{q_1}{q_2}\left(\left(2d+1\right)\left(1-q_2\right)-2d\ln\left(\frac{1}{q_2}\right)\right).\end{aligned}$$

The first inequality is due to Lemma 2 and the second inequality follows from the fact that $\sum_{k=q_2n+1}^{n} \frac{1}{k} \leq \int_{q_2n}^{n} \frac{1}{x}dx = \ln\left(\frac{1}{q_2}\right)$. Overall we get

$$\begin{aligned}\mathrm{E}\left[\mathrm{ALG}\right] \geq \mathrm{OPT}\left(\mathcal{I}_{heavy}\right) \frac{q_1}{d} \ln\left(\frac{q_2}{q_1}\right) \\ + \mathrm{OPT}\left(\mathcal{I}_{light}\right) \frac{q_1}{q_2}\left(\left(2d+1\right)\left(1-q_2\right) - 2d\ln\left(\frac{1}{q_2}\right)\right). \end{aligned} \tag{1}$$

For the parameters $q_2 = 2d/\left(2d+1\right)$, $q_1 = q_2/\sqrt[4]{e}$, we have

$$\frac{q_1}{d}\ln\left(\frac{q_2}{q_1}\right) = \frac{1}{4\sqrt[4]{e}}\frac{2}{2d+1} = \frac{1}{\sqrt[4]{e}(4d+2)}.$$

Using the fact that for $x \geq 0$, $\ln\left(1+x\right) \leq x - \frac{1}{2}x^2 + \frac{1}{3}x^3$, we have

$$\frac{q_1}{q_2}\left(\left(2d+1\right)\left(1-q_2\right) - 2d\ln\left(\frac{1}{q_2}\right)\right) = \frac{1}{\sqrt[4]{e}}\left(1 - 2d\ln\left(1+\frac{1}{2d}\right)\right) \geq \frac{1}{\sqrt[4]{e}}\left(\frac{1}{4d} - \frac{1}{12d^2}\right).$$

It can be easily verified that $\left(1/4d - 1/12d^2\right) \geq 1/(4d{+}2)$ for $d \geq 1$. Now since $\mathrm{OPT}\left(\mathcal{I}_{heavy}\right) + \mathrm{OPT}\left(\mathcal{I}_{light}\right) \geq \mathrm{OPT}\left(\mathcal{I}\right)$, we get

$$\mathrm{E}\left[\mathrm{ALG}\right] \geq \frac{1}{\sqrt[4]{e}(4d+2)}\left(\mathrm{OPT}\left(\mathcal{I}_{heavy}\right) + \mathrm{OPT}\left(\mathcal{I}_{light}\right)\right) \geq \frac{1}{\sqrt[4]{e}(4d+2)}\mathrm{OPT}. \qquad \blacktriangleleft$$

It is important to note that for $d = 1$, the competitive-ratio can be improved by choosing $q_1 = 0.5256$ and $q_2 = 0.69$. Setting these parameters in (1) shows that Algorithm 1 is 6.99-competitive for the (one-dimensional) generalized assignment problem, which improves upon the best-known competitive-ratio of 8.1 achieved by Kesselheim et al. [14].

▶ Remark 4. Algorithm 1 can easily be extended to the case where each item has $K \geq 1$ different packing options in each bin in a similar way to the algorithm of Kesselheim et al. [14]. Therefore, the general online packing LPs problem with n requests and m resources can be viewed as a special case of VGAP with one m-dimensional bin and n items.

3.1 The {0,1}-VGAP

The $\{0,1\}$-*VGAP* is a special case of VGAP in which the consumption of item i from bin j in dimension t is either 0 or the whole capacity of bin j in dimension t. By scaling, we can assume without loss of generality that $\mathbf{b}_j = \mathbf{1}$ for all $j \in [m]$, and $\mathbf{w}_{i,j} \in \{0,1\}^d$, $\forall i \in [n]$, $\forall j \in [m]$.[2] Note that for $d = 1$ the problem is identical to weighted bipartite matching.

As for the general VGAP, given an instance $\mathcal{I}$ we partition it into two sub-instances, however, we make the partition according to the density of the weight vectors: we call the packing option of item i in bin j *dense* if $|\mathrm{supp}\left(\mathbf{w}_{i,j}\right)| \geq \sqrt{d}$,otherwise, we call it *sparse*.[3] We denote by $\mathcal{I}_{dense}$ the sub-instance that consists only of the dense packing options of every item, and by $\mathcal{I}_{sparse}$ the complementary sub-instance that consists only of the sparse packing options.

Our algorithm for this case, which we call Algorithm 3, is based on the simple observation that in $\mathcal{I}_{dense}$ at most $\sqrt{d}$ items can be packed in every bin, therefore, a maximum weight matching in $G(\mathcal{I}_{dense})$ has a weight of at least $\mathrm{OPT}(\mathcal{I}_{dense})/\sqrt{d}$. Algorithm 3 is almost

[2] $\mathbf{1}$ denotes the all 1's vector.
[3] $\mathrm{supp}(\cdot)$ denotes the set of indices of non-zero entries of a vector.

identical to Algorithm 1, the only difference is that $\mathcal{I}_{heavy}$ and $\mathcal{I}_{light}$ are replaced with $\mathcal{I}_{dense}$ and $\mathcal{I}_{sparse}$ respectively (for a full description see Appendix A.1). The parameters q_1 and q_2 are defined in the analysis. Due to lack of space, we only state the result of our analysis in Theorem 5 and give the full proof in Appendix A.2.

▶ **Theorem 5.** *For* $q_2 = \sqrt{d}/\left(\sqrt{d}+1\right)$ *and* $q_1 = q_2/\sqrt{e}$*, Algorithm 3 is* $2\sqrt{e}\left(\sqrt{d}+2\right)$*-competitive.*

4 Vector Multiple Knapsack Problem

The *Vector Multiple Knapsack Problem* (VMKP) is a special case of VGAP in which all bins have a capacity of $\mathbf{1}$, every packing option of item i consumes the same amount of capacity $\mathbf{w}_i \in [0,1]^d$ and provides the same profit $p_i \geq 0$, i.e., $\mathbf{w}_{i,j} = \mathbf{w}_i$, $p_{i,j} = p_i$, $\forall i \in [n]$, $\forall j \in [m]$. We study the case where there are at least two bins, i.e., $m \geq 2$. For this special case we present an online algorithm that improves upon the competitive-ratio of Algorithm 1.

Algorithm 2 Online VMKP.

$S_0 \leftarrow \emptyset$, $\mathrm{P}_0 \leftarrow \emptyset$;
for *each item* i_ℓ *that arrives at round* ℓ **do**
  $S_\ell \leftarrow S_{\ell-1} \cup \{i_\ell\}$;
  if $\ell \leq qn$ **then** `/* sampling phase */`
    continue to the next round;
  else `//` $\ell \geq qn+1$ `/* packing phase */`
    Let $x^{(\ell)}$ be an optimal fractional solution for the LP-relaxation of $\mathcal{I}|_{S_\ell}$;
    Choose j_ℓ randomly where $\Pr[j_\ell = j] = x^{(\ell)}_{i_\ell,j}$ and $\Pr[j_\ell = 0] = 1 - \sum_{j\in[m]} x^{(\ell)}_{i_\ell,j}$;
    `// First Fit`
    Let $B_\ell = \{j \in [m] : \mathrm{P}_\ell \cup \{(i_\ell, j)\} \text{ is feasible}\}$;
    if $j_\ell \neq 0$ ***and*** $B_\ell \neq \emptyset$ **then**
      $\mathrm{P}_\ell \leftarrow \mathrm{P}_{\ell-1} \cup \{(i_\ell, \min B_\ell)\}$;
return P_n

Algorithm 2 consists of two phases: a sampling phase and a packing phase. The packing phase is similar to the light phase of Algorithm 1, however, instead of using the LP-solution to compute a tentative assignment, it uses it only to make a binary decision whether to pack the current item or not. Still, we keep a randomized rounding step similar to Algorithm 1 in order to use observations made in Section 3. For the actual packing, Algorithm 2 exploits the fact that all packing options are identical and uses the First Fit algorithm [12].

We now analyze the performance of Algorithm 2. First we prove a simple observation due to the nature of First Fit.

▶ **Lemma 6.** *For* $\ell \geq qn+1$ *and* $m \geq 2$*, if* i_ℓ *cannot be packed in any bin, then* $\sum_{(i,j)\in \mathrm{P}_{\ell-1}} \sum_{t=1}^{d} w_i^t \geq m/2$.

Proof. Let $u(j,t,\ell)$ denote the total consumption of bin j in dimension t before round ℓ. Since i_ℓ cannot be packed in any bin, there is at least one item packed in each bin. Consider any two bins $j' > j$, and let i_k be the first item that was packed in j'. i_k could not be packed in bin j, therefore, for some $t' \in [d]$ we have $u(j,t',k) + w_{i_k}^{t'} > 1$. Since

the consumption is non-decreasing and $u(j', t', \ell) \geq w_{i_k}^{t'}$ we have $u(j, t', \ell) + u(j', t', \ell) > 1$, therefore, $\sum_{t=1}^{d} u(j, t, \ell) + u(j', t, \ell) > 1$. By summing the last inequality for all consecutive pairs of bins $(j+1, j)$ as well as $(m, 1)$ we get $2\sum_{j=1}^{m}\sum_{t=1}^{d} u(j, t, \ell) > m$ and hence the lemma. ◀

Next, we follow the method of the previous section to bound the expected profit of the algorithm at each round.

▶ **Lemma 7.** *For $\ell \geq qn+1$ and $m \geq 2$, we have* $\mathrm{E}[R_\ell] \geq \left(1 - 2d\sum_{k=qn+1}^{\ell-1} \frac{1}{k}\right)\frac{1}{n}\mathrm{OPT}$.

Proof. By following a similar argument to that in Lemma 1, we get $\mathrm{E}[p_{i_\ell, j_\ell}] \geq \frac{1}{n}\mathrm{OPT}$ for $\ell \geq qn+1$. We now bound the probability that $\sum_{(i,j)\in P_{\ell-1}}\sum_{t=1}^{d} w_i^t < m/2$, by Lemma 6, this is a sufficient condition for the assignment of i_ℓ to be successful. At round $k < \ell$ the algorithm computes a tentative assignment based on an optimal fractional solution $x^{(k)}$ for the LP-relaxation of $\mathcal{I}|_{S_k}$, therefore, we have $\sum_{i\in S_k}\sum_{t=1}^{d}\sum_{j=1}^{m} x_{i,j}^{(k)} w_i^t \leq dm$. Since i_k is a uniformly random item of S_k, we have $\mathrm{E}\left[\sum_{t=1}^{d}\sum_{j=1}^{m} x_{i_k,j}^{(k)} w_{i_k}^t\right] \leq dm/k$, hence,

$$\begin{aligned}\mathrm{E}\left[\sum_{(i,j)\in P_{\ell-1}}\sum_{t=1}^{d} w_i^t\right] &\leq \mathrm{E}\left[\sum_{k=qn+1}^{\ell-1}\sum_{t=1}^{d} w_{i_k}^t \sum_{j=1}^{m} x_{i_k,j}^{(k)}\right] \\ &= \sum_{k=qn+1}^{\ell-1} \mathrm{E}\left[\sum_{t=1}^{d}\sum_{j=1}^{m} x_{i_k,j}^{(k)} w_{i_k}^t\right] \leq \sum_{k=qn+1}^{\ell-1}\frac{dm}{k}.\end{aligned}$$

As before, we can now use Markov's inequality to bound the probability of successful assignment and get the lemma. ◀

▶ **Theorem 8.** *For $q = 2d/(2d+1)$, Algorithm 2 is $(4d+2)$-competitive.*

Proof. By Lemma 7, the overall profit of the algorithm is bounded by

$$\mathrm{E}[\mathrm{ALG}] \geq \sum_{\ell=qn+1}^{n}\left(1 - 2d\sum_{k=qn+1}^{\ell-1}\frac{1}{k}\right)\frac{1}{n}\mathrm{OPT} \geq \left((2d+1)(1-q) - 2d\ln\left(\frac{1}{q}\right)\right)\mathrm{OPT}.$$

This bound is maximized for $q = 2d/(2d+1)$, and for this choice of parameter, using similar arguments as in the proof of Theorem 3, we get

$$\begin{aligned}\mathrm{E}[\mathrm{ALG}] &\geq \left(1 - 2d\ln\left(1 + \frac{1}{2d}\right)\right)\mathrm{OPT} \qquad (2)\\ &\geq \left(\frac{1}{4d} + \frac{1}{12d^2}\right)\mathrm{OPT} \geq \left(\frac{1}{4d+2}\right)\mathrm{OPT}.\end{aligned}$$

◀

Note that by setting $d = 1$ in (2) we get that Algorithm 2 is 5.29-competitive for the (one-dimensional) multiple knapsack problem with at least two bins.

▶ Remark 9. For the special case of $d = 1$, Algorithm 2 can be implemented in a more efficient way: instead of solving an LP-relaxation at every round of the packing phase, we can obtain an optimal fractional solution by using a simple greedy algorithm.

▶ Remark 10. Algorithm 2 can be extended to the case of variable-sized squared bins, that it, to the case where $\mathbf{b}_j = \mathbf{1} \cdot b_j$, $\forall j \in [m]$, under the assumption that every item fits into every bin, i.e., $w_i^t \leq b_j \ \forall i \in [n], \forall j \in [m], \forall t \in [d]$, through sorting the bins by their capacity in a non-increasing order.

5 Lower bound

We now prove a lower bound of $\Omega(d)$ for the vector knapsack problem (VMKP with a single bin). Since it is a special case of VGAP, it shows our $O(d)$-competitive algorithm for VGAP from Section 3 is asymptotically optimal. Note that our lower bound holds without any complexity assumptions. In particular, it also applies to algorithms with unbounded computational power. The proof is inspired by the work of Babaioff et al. [5].

We construct an instance for the d-dimensional knapsack problem consisting of one bin of capacity $\mathbf{1}$ and $n = \delta d^{(\delta+1)d+1}$ items, where $\delta \in \mathbb{N}_+$. The weight vectors of the items are the columns of the following $d \times d$ matrices:

$$A_j = \left(1 - \epsilon j d^j\right) \cdot I + \epsilon j d^{j-1} \cdot \left(\mathbf{1}\mathbf{1}^T - I\right), \quad \forall j \in [\delta d^{(\delta+1)d}].$$

Where I is the $d \times d$ identity matrix, and $\epsilon < 1/\left(2nd^n\right)$. By the choice of ϵ it holds that $\epsilon j d^j < 1/2$, $\forall j \in [\delta d^{(\delta+1)d}]$.

Observe that for every matrix A_j, all the items that correspond to its columns fit together in the bin, that is, $A_j \cdot \mathbf{1} \leq \mathbf{1}$. Also, every two columns of different matrices cannot be packed together. This is true because for any two matrices A_i, A_j where $i > j$, and any two columns $k, \ell \in [d]$, we have

$$(A_j)_{k,k} + (A_i)_{k,\ell} \geq \left(1 - \epsilon j d^j\right) + \epsilon i d^{i-1} \geq 1 - \epsilon j d^j + \epsilon \left(j+1\right) d^j > 1.$$

The first inequality follows from the fact that $\epsilon i d^{i-1} < \left(1 - \epsilon i d^i\right)$, and the second inequality follows from the fact that $i \geq j+1$. Every item is independently assigned a profit of 1 with probability $1/d^{\delta+1}$ and 0 with probability $1 - 1/d^{\delta+1}$.

▶ **Theorem 11.** *Any online algorithm produces a packing with expected profit of at most* $\left(1 + \frac{1}{d^\delta}\right)$*, while* $\mathrm{OPT} = d$ *with probability of at least* $\left(1 - \frac{1}{e^\delta}\right)$.

Proof. Let us observe the first item that the online algorithm packs. It corresponds to a column of one matrix A_j. All items that correspond to columns of different matrices cannot be packed along with it. The only items that can be added to the packing are the remaining columns of A_j. There are less than d such items left, each has an expected profit of $1/d^{\delta+1}$. Since the first item has a profit of at most 1, the expected profit of the packing produced by the algorithm is at most $1 + 1/d^\delta$.

With regard to the optimal packing, for a given matrix A_ℓ, the probability that all items are of profit 1 is $1/d^{(\delta+1)d}$, therefore, the probability that all matrices are of weight less than d is $\left(1 - 1/d^{(\delta+1)d}\right)^{d^{(\delta+1)d}\delta} \leq 1/e^\delta$. ◀

Note that Theorem 11 can be easily modified to apply to the case of two identical bins, thus, it shows that Algorithm 2 for VMKP with at least two bins is also asymptotically optimal.

6 Conclusions

In this paper, we presented simple, asymptotically optimal, online algorithms for multidimensional variants of the generalized assignment problem in the random-order model, which has vast implications for real-world applications, like resource allocation in cloud computing.

Our bounds for VGAP are translated to a matching lower and upper bounds of $\Omega(m)$ and $O(m)$ for the general online packing LPs problem (as mentioned in Remark 4, where m is the number of resources).

For the one-dimensional case, the best lower bound for the online GAP is derived from the lower bound for the secretary problem of e [6]. An interesting open question is to close the gap between e and the upper bound of 6.99 presented in this paper. It is also very interesting to understand whether the new theoretical algorithm provides practical value for cloud resource allocation, where the value of d is a small constant (2 or 3).

References

1 Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online Vertex-Weighted Bipartite Matching and Single-bid Budgeted Allocations. In *SODA*, pages 1253–1264. SIAM, 2011.

2 Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A Dynamic Near-Optimal Algorithm for Online Linear Programming. *Operations Research*, 62(4):876–890, 2014.

3 Yossi Azar, Ilan Reuven Cohen, Seny Kamara, and F. Bruce Shepherd. Tight bounds for online vector bin packing. In *STOC*, pages 961–970. ACM, 2013.

4 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A Knapsack Secretary Problem with Applications. In *APPROX-RANDOM*, volume 4627 of *Lecture Notes in Computer Science*, pages 16–28. Springer, 2007.

5 Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443. SIAM, 2007.

6 Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary Problems via Linear Programming. *Math. Oper. Res.*, 39(1):190–206, 2014.

7 Chandra Chekuri and Sanjeev Khanna. On Multi-Dimensional Packing Problems. In *SODA*, pages 185–194. ACM/SIAM, 1999.

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

9 Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *SODA*, pages 395–404. SIAM, 2005.

10 Eugene B Dynkin. The optimum choice of the instant for stopping a Markov process. *Soviet Mathematics*, 4:627–629, 1963.

11 Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Clifford Stein. Online Stochastic Packing Applied to Display Ad Allocation. In *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 182–194. Springer, 2010.

12 Michael R Garey, Ronald L Graham, David S Johnson, and Andrew Chi-Chih Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory, Series A*, 21(3):257–298, 1976.

13 Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions. In *ESA*, volume 8125 of *Lecture Notes in Computer Science*, pages 589–600. Springer, 2013.

14 Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal Beats Dual on Online Packing LPs in the Random-Order Model. *SIAM J. Comput.*, 47(5):1939–1964, 2018.

15 Robert D. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA*, pages 630–631. SIAM, 2005.

16 Benny Lehmann, Daniel J. Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

17 Denis V Lindley. Dynamic programming and decision theory. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 10(1):39–51, 1961.

18 Aranyak Mehta, Amin Saberi, Umesh V. Vazirani, and Vijay V. Vazirani. AdWords and generalized online matching. *J. ACM*, 54(5):22, 2007.

19 Marco Molinaro and R. Ravi. Geometry of Online Packing Linear Programs. In *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 701–713. Springer, 2012.

20 Danny Raz, Itai Segall, and Maayan Goldstein. Multidimensional resource allocation in practice. In *Proceedings of the 10th ACM International Systems and Storage Conference*, page 1. ACM, 2017.

21 Lei Shi, Bernard Butler, Dmitri Botvich, and Brendan Jennings. Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 499–505. IEEE, 2013.

22 David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.

A Omitted Details

A.1 Description of Algorithm 3

Algorithm 3 Online $\{0,1\}$-VGAP.

$S_0 \leftarrow \emptyset$, $\mathrm{P}_0 \leftarrow \emptyset$;
for *each item* i_ℓ *that arrives at round* ℓ **do**
  $S_\ell \leftarrow S_{\ell-1} \cup \{i_\ell\}$;
  if $\ell \leq q_1 n$ **then** `/* sampling phase */`
    continue to the next round;
  else if $q_1 n + 1 \leq \ell \leq q_2 n$ **then** `/* dense phase */`
    Let $x^{(\ell)}$ be a maximum-weight matching in $G\left(\mathcal{I}_{dense}|_{S_\ell}\right)$;
    `// compute a tentative assignment` (i_ℓ, j_ℓ)
    if i_ℓ *is matched in* $x^{(\ell)}$ **then**
      Let j_ℓ be the bin to which i_ℓ is matched;
    else
      $j_\ell \leftarrow 0$
    if $j_\ell \neq 0$ ***and*** j_ℓ *is empty in* $\mathrm{P}_{\ell-1}$ **then**
      $\mathrm{P}_\ell \leftarrow \mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$;
  else `//` $(\ell \geq q_2 n + 1)$ `/* sparse phase */`
    Let $x^{(\ell)}$ be an optimal fractional solution for the LP-relaxation of $\mathcal{I}_{sparse}|_{S_\ell}$;
    `// compute a tentative assignment` (i_ℓ, j_ℓ) `by randomized rounding`
    Choose bin j_ℓ randomly where $\Pr[j_\ell = j] = x^{(\ell)}_{i_\ell,j}$ and
    $\Pr[j_\ell = 0] = 1 - \sum_{j \in [m]} x^{(\ell)}_{i_\ell,j}$;
    if $j_\ell \neq 0$ ***and*** $\mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$ *is feasible* **then**
      $\mathrm{P}_\ell \leftarrow \mathrm{P}_{\ell-1} \cup \{(i_\ell, j_\ell)\}$;
  return P_n

A.2 Proof of Theorem 5

Proof. To prove the competitive-ratio of Algorithm 3, we bound the expected profit pf the algorithm in round separately. In Lemma 12 below, we bound the expected profit in each round of the dense phase, and in Lemma 13, we bound the expected profit in each round of the sparse phase. Then we sum over the expected profit in both phases to get the total profit of the algorithm.

▶ **Lemma 12.** *For $q_1 n + 1 \leq \ell \leq q_2 n$, we have* $\mathrm{E}[R_\ell] \geq \frac{q_1}{\ell - 1} \cdot \frac{1}{\sqrt{d}} \mathrm{OPT}(\mathcal{I}_{dense})$.

The proof is similar to the proof of Lemma 1 by using the observation that in $\mathcal{I}_{dense}$ at most $\sqrt{d}$ items can be packed in every bin, therefore, a maximum weight matching in $G(\mathcal{I}_{dense})$ has a weight of at least $\mathrm{OPT}(\mathcal{I}_{dense})/\sqrt{d}$. .

▶ **Lemma 13.** *For $q_1 n + 1 \leq \ell \leq q_2 n$, we have*

$$\mathrm{E}\left[R_\ell\right] \geq \frac{q_1}{q_2} \left(1 - \sqrt{d} \sum_{k=q_2 n+1}^{\ell-1} \frac{1}{k}\right) \frac{1}{n} \mathrm{OPT}\left(\mathcal{I}_{sparse}\right).$$

Proof. As in Lemma 2, we have $\mathrm{E}\left[p_{i_\ell, j_\ell}\right] = \frac{1}{n}\mathrm{OPT}\left(\mathcal{I}_{sparse}\right)$ where the expectation is taken only over the random choice of the subset $S_\ell \subseteq [n]$, the random choice of $i_\ell \in S_\ell$ and the internal randomness of the algorithm at round ℓ. Once again we bound the probability of successful assignment over the random arrival order of previous items and the internal randomness of the algorithm in previous rounds. The assignment of i_ℓ to j_ℓ must be successful if the following conditions hold: (1) no item was packed in j_ℓ during the dense phase, and (2) no tentative assignments from previous rounds of the sparse phase occupy the entries in $\mathrm{supp}\left(\mathbf{w}_{i_\ell, j_\ell}\right)$ of j_ℓ. Let us denote event (1) by H_ℓ and the event described in (2) by L_ℓ. At round $q_2 n \leq k \leq \ell$ the algorithm uses an optimal fractional solution $x^{(k)}$ for the LP-relaxation on $\mathcal{I}|_{S_k}$ to compute a tentative assignment (i_k, j_k). In that solution we have $\sum_{i \in S_k} x^{(k)}_{i,j_\ell} w^t_{i,j_\ell} \leq 1$, $\forall t \in [d]$. Observe that by the randomized rounding at round k and the fact that $w^t_{i_k, j_\ell} \in \{0, 1\}$, the probability that the tentative assignment of i_k uses dimension t in j_ℓ is given by $\sum_{i \in S_k} \Pr[j_k = j_\ell \wedge w^t_{i,j_k} = 1 | i_k = i] \cdot \Pr[i_k = i] = \frac{1}{k} \sum_{i \in S_k} x^{(k)}_{i,j_\ell} w^t_{i,j_\ell} \leq \frac{1}{k}$. Using a union bound, since $|\mathrm{supp}\left(\mathbf{w}_{i_\ell, j_\ell}\right)| \leq \sqrt{d}$, the probability that i_k blocks i_ℓ from being packed is at most $\sqrt{d}/k$. Applying a union bound once again over all previous rounds of the sparse phase, we get $\Pr[L_\ell] \geq 1 - \sum_{k=q_2 n+1}^{\ell-1} \sqrt{d}/k$. From here on we can follow a similar argument as in the proof of Lemma 2 and get that the probability of successful assignment is at least

$$\Pr\left[H_\ell \wedge L_\ell\right] \geq \prod_{k=q_1 n+1}^{q_2 n} \left(1 - \frac{1}{k}\right) \left(1 - \sum_{k=q_2 n+1}^{\ell-1} \frac{\sqrt{d}}{k}\right) = \frac{q_1}{q_2} \left(1 - \sqrt{d} \sum_{k=q_2 n+1}^{\ell-1} \frac{1}{k}\right).$$

Overall, we get the lemma. ◀

By using Lemma 12 and Lemma 13 to sum over the profit raised in each phase, we get

$$\begin{aligned}\mathrm{E}\left[\mathrm{ALG}\right] \geq\ & \mathrm{OPT}\left(\mathcal{I}_{dense}\right) \frac{q_1}{\sqrt{d}} \ln\left(\frac{q_2}{q_1}\right) \\ & + \mathrm{OPT}\left(\mathcal{I}_{sparse}\right) \frac{q_1}{q_2} \left(\left(\sqrt{d} + 1\right)(1 - q_2) - \sqrt{d} \ln\left(\frac{1}{q_2}\right)\right).\end{aligned}$$

Setting $q_2 = \frac{\sqrt{d}}{\sqrt{d}+1}$, $q_1 = q_2/\sqrt{e}$, we get

$$\begin{aligned}\mathrm{E}\left[\mathrm{ALG}\right] \geq\ & \mathrm{OPT}\left(\mathcal{I}_{dense}\right) \frac{1}{2\sqrt{e}\left(\sqrt{d} + 1\right)} \\ & + \mathrm{OPT}\left(\mathcal{I}_{sparse}\right) \frac{1}{\sqrt{e}} \left(1 - \sqrt{d} \ln\left(1 + \frac{1}{\sqrt{d}}\right)\right). \qquad (3)\end{aligned}$$

To bound the second term we use the fact that for $x \geq 0$, $\ln(1+x) \leq x - \frac{1}{2}x^2 + \frac{1}{3}x^3$ and get

$$\frac{1}{\sqrt{e}}\left(1 - \sqrt{d}\ln\left(1+\frac{1}{\sqrt{d}}\right)\right) \geq \frac{1}{\sqrt{e}}\left(\frac{1}{2\sqrt{d}} - \frac{1}{3d}\right) \geq \frac{1}{2\sqrt{e}\left(\sqrt{d}+1\right)}. \tag{4}$$

The second inequality can be easily verified to holds for $d \geq 1$. By substituting (4) in Equation (3) and using the fact that $\mathrm{OPT}\left(\mathcal{I}_{dense}\right) + \mathrm{OPT}\left(\mathcal{I}_{sparse}\right) \geq \mathrm{OPT}\left(\mathcal{I}\right)$, we get the theorem. ◀