Approximability of Packing and Covering Problems

A THESIS
SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
Master of Technology
IN
Faculty of Engineering

BY

Arka Ray

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

February, 2022

Declaration of Originality

I, **Arka Ray**, with SR No. **04-04-00-10-42-19-1-16844** hereby declare that the material presented in the thesis titled

Approximability of Packing and Covering Problems

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2019-2021**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the thesis.

Advisor Name: Arindam Khan Advisor Signature

DEDICATED TO

Ma and Baba

for believing in me

Acknowledgements

My sincerest gratitude to Arindam Khan, who, despite the gloom and doom situation of the present times, kept me motivated through his constant encouragement while tolerating my idiosyncrasies. I would also like to thank him for introducing me to approximation algorithms, along with Anand Louis, in general, and approximation of packing and covering problems in particular.

I would like to thank Rameesh Paul, KVN Sreenivas, and Eklavya Sharma for their valuable comments.

Abstract

Packing and covering problems form a large and important class of problems in computer science. Many packing and covering problems are known to be NP-hard and hence we study them in the context of approximation algorithms.

In this thesis, we look at vector bin packing, and vector bin covering which are multidimensional extensions of the bin packing problem and bin covering problems, respectively. In the vector bin packing problem given a set of vectors S from $(0,1]^d$, the aim is to obtain a minimum cardinality partition of S into bins $\{B_i\}$ such that for each B_i, we have $\left\|\sum_{v \in B_i} v\right\|_\infty \leq 1$. Woeginger [44] claimed that the vector bin packing has no APTAS. We note a minor oversight in his proof and revise it to show that there is no algorithm for vector bin packing with an asymptotic approximation ratio better than $\frac{600}{599}$ unless P = NP. Vector bin covering is the covering analogue of the vector bin packing problem where given a set of vectors S from $(0,1]^d$, the aim is to obtain a disjoint family of subsets (also called bins) with the maximum cardinality such that for each bin B, we have $\sum_{v \in B} v \geq \mathbf{1}$. We also show that is not possible to obtain an algorithm with an asymptotic approximation ratio better than $\frac{998}{997}$ unless P = NP.

We also study the multidimensional extensions of min-knapsack problem, which is the covering variant of the knapsack problem. For vector min-knapsack we obtain a PTAS and a matching lower bound showing that there is no EPTAS unless W[1]=FPT. In case of the geometric min-knapsack we show that there is no algorithm which can decide if there is a feasible solution to a given instance hence showing that there is no polynomial-time approximation algorithm possible for it.

Publications based on this Thesis

1. Arka Ray. There is no APTAS for 2-dimensional vector bin packing: Revisited. *CoRR*, abs/2104.13362, 2021. URL https://arxiv.org/abs/2104.13362.

Contents

List of Figures

Chapter 1

Introduction

I get ideas about what's essential when packing my suitcase.
— *Diane von Furstenberg*

1.1 Packing and Covering Problems

Packing and Covering problems are two of the most important classes of optimization problems in computer science. Many of the problems in Karp's 21 NP-complete problems in [34] are in fact packing and covering problems.

Roughly speaking, in a packing problem we have some notion of a container and items which need to be packed into these containers. The objective is to either pack items into a single container so as to maximize some quantity associated with the items or to pack all the items into minimum number of containers possible. The archetypical example of the problem of the first kind is *the knapsack problem* in which given a set of items I with each item i having size $s_i \in (0, 1]$ and profits p_i, the aim is to find a subset I' such that $\sum_{i \in I'} s_i \leq 1$ and $\sum_{i \in I'} p_i$ is maximized. While the quintessential example of a problem of the second kind is *the bin packing problem* in which given a set of items I with each item i having size $s_i \in (0, 1]$, the aim is to find a minimum cardinality partition of the items into subsets $\{B_i\}$ called bins such that for each bin $\sum_{j \in B_i} s_j \leq 1$. In context of bin packing any subset $B \subseteq I$ is also called a *configuration* of a bin or simply configuration. Other examples of packing problems include *the set packing problem* in which given a ground set $E = \{e_i | i \in [n]\}$, a collection of subsets $\{S_1, S_2, \ldots, S_m\}$, and w_i associated with each such subset S_j, the aim is find a set of indices $I \subseteq [m]$ which maximizes $\sum_{j \in I} w_i$ while $S_j \cap S_{j'} = \emptyset$ for every $j, j' \in I$; *the independent set problem* in which given a graph $G = (V, E)$, the aim is to find the maximum cardinality subset $I \subseteq V$ such that for any $v, w \in I$, $\{v, w\} \notin E$; *the vertex coloring problem* in which given a graph $G = (V, E)$, the aim is to partition the graph into minimum cardinality family of subsets $\{C_1, \ldots, C_m\}$ of V called colors such that for each color C_i, $v, w \in C_i$ implies $\{v, w\} \notin E$.

In a similar vein, there is a notion of covering some object using some items. Typically in covering problems, the objective is either to cover an object with items with least cost or to cover as many objects as possible. One typical example of problem of the first kind of covering problem is *the set cover problem* where given a ground set $E = \{e_i | i \in [n]\}$, a family of subsets $\{S_1, \ldots, S_m\}$, and weights w_i associated with each set S_i, the aim is to obtain a set of indices $I \subseteq [m]$ such that $\bigcup_{i \in I} S_i = E$ while minimizing $\sum_{i \in I} w_i$. A representative problem for the second kind of problem is the bin covering problem where given a set of items I with sizes $s_i \in (0, 1]$, the aim is obtain a minimum cardinality family of disjoint subsets $\mathcal{F} = \{B_i\}$ of I such that for each $B_i \in \mathcal{F}$, we have $\sum_{j \in B_i} s_j \geq 1$. Each member of this family can be referred to as a bin or a unit cover. Other examples of covering problems include *the vertex cover problem* where given a graph $G = (V, E)$, the aim is to determine the minimum cardinality subset $C \subseteq V$ such that for each edge $\{v, w\} \in E$ either $v \in C$ or $w \in C$; *the edge cover problem* where given a graph $G = (V, E)$, the aim is to determine the minimum cardinality subset of edges F such that for every vertex v there is a $w \in V$ such that $\{v, w\} \in F$. Note that many of these problems are dual of some packing problem.

Now, as we noted earlier many of the important packing and covering problems are known to be NP-complete. So, the best we can hope is to get an approximation algorithm[1] for these problems. Some of these problems like the knapsack problem admit a FPTAS (see [27, 37]), while others like bin packing do not have PTAS (folklore) but admit an APTAS, whereas yet others like the independent set problem and vertex coloring do not even admit a constant approximation.

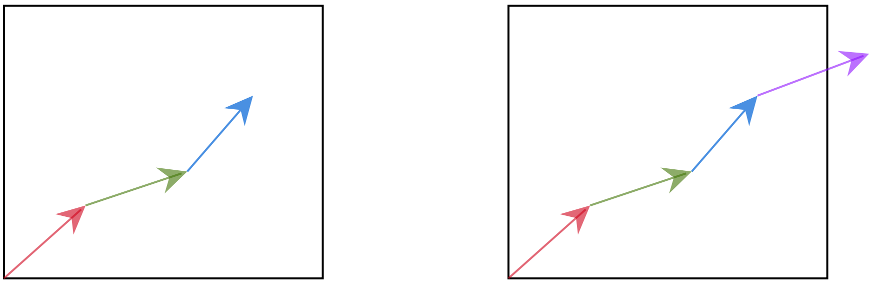
Therefore, in this thesis we have studied the approximability of multidimensional variants of bin packing, bin covering, and min-knapsack (a covering variant of knapsack, see Chapter 4). We now present a brief survey of existing work on bin packing, bin covering, knapsack and a few variants thereof.

1.2 Related Works

1.2.1 Bin Packing

Garey et al. [23] were the first to study the bin packing problem. They studied it in the context of memory allocation and gave some bounds on the asymptotic approximation ratios for first-fit, best-fit, first-fit decreasing, and best-fit decreasing. There is a well-know reduction from

[1] see Chapter 2 for definition of approximation ratio and related notions



(a) Valid Configuration (b) Invalid Configuration

Figure 1.1: configurations for 2-d vector bin packing

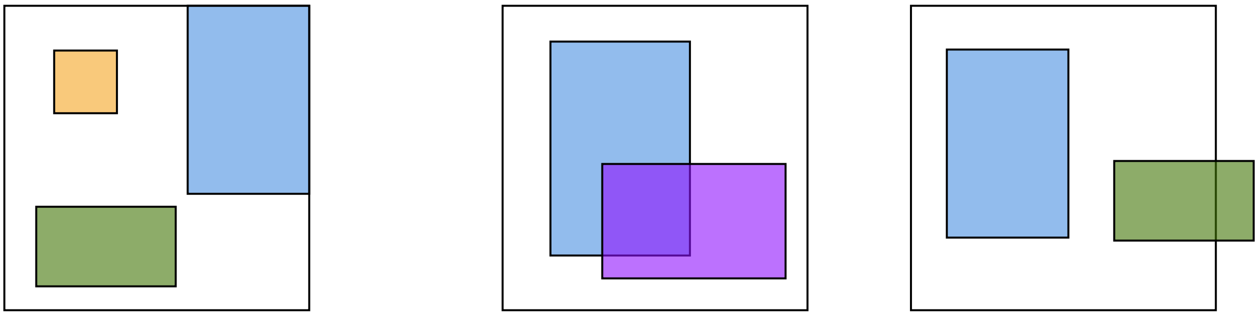
the partition problem to the bin packing problem which not only shows that bin packing is NP-complete but also show that (absolute) approximation ratio better than $3/2$ is not possible unless P=NP. For this reason the focus of bin packing research has been on getting a better asymptotic approximation ratio.

de la Vega and Lueker [19] introduced the linear grouping technique, where number of types of items is reduced by rounding up the sizes of the items, using which they gave an APTAS. This technique has led to improved algorithms for many related problems. This bound was further improved in a series of work by Karp and Karmarkar [33], Rothvoß [40], and Hoberg and Rothvoß [26] to $\mathrm{OPT} + O(\log^2 \mathrm{OPT})$, $\mathrm{OPT} + O(\log \mathrm{OPT} \log\log \mathrm{OPT})$, and $\mathrm{OPT} + O(\log \mathrm{OPT})$, respectively. As the reduction shown above does not even preclude an $\mathrm{OPT} + 1$, therefore it is an open problem if such an algorithm exists or not.

Two multidimensional variants of bin packing which have been studied in the literature: (i) vector bin packing (VBP) and (ii) geometric bin packing (GBP). In the vector bin packing problem given a set of vectors S from $(0, 1]^d$, the aim is to obtain a minimum cardinality partition of S into bins $\{B_i\}$ such that for each B_i, we have $\left\| \sum_{v \in B_i} v \right\|_\infty \leq 1$. As is the case with many multidimensional extension of problems, we can study vector bin packing under two regimes: (i) the case where the dimension is part of the input, (ii) the case where the dimension is a fixed constant.

In the case where the dimension d has been supplied as part of the input de la Vega and Lueker [19] gave a $(d + \epsilon)$ approximation. Bansal et al. [6] discuss a reduction from the vertex coloring problem to vector bin packing problem with arbitrary dimensions pointed to them by Jan Vondrák. This reduction shows that there is no $d^{1-\epsilon}$ approximation for any $\epsilon > 0$ unless NP=ZPP.

If d is kept constant, i.e., it is not supplied as part of the input then the above bound does not hold, and in fact much better results are known for this case. The first result breaking the



(a) valid configuration (b) invalid configurations

Figure 1.2: configurations of 2-d dimensional geometric bin packing

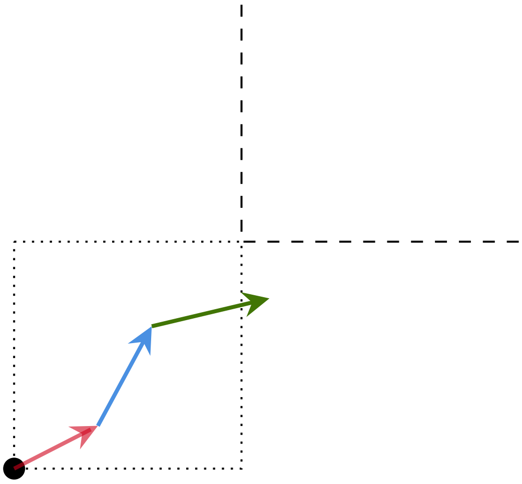
barrier of d was the $1+\epsilon d+H_{\epsilon^{-1}}$ by Chekuri and Khanna [10] where $H_k=\sum_{m=1}^{k}\frac{1}{m}$. Notice that taking $\epsilon=\frac{1}{d}$, this implies an $O(\ln d)$ approximation (in fact $\ln d+2+\gamma$[1]). This was further improved to $\ln d+1$ by Bansal, Caprara and Sviridenko [5] and then to $\ln(d+1)+0.807$ by Bansal, Eliás, and Khan [6] using the Round and Approx framework. For the $d=2$ case these translate to $1+\ln 2\approx 1.693$ and $1+\ln(1.5)\approx 1.406$. Finally, Sandeep [41] gave an $\Omega(\ln d)$ lower bound to match the above algorithms.

Geometric bin packing is in fact a class of problems. In geometric bin packing problems the bins and the items are replaced by some geometric object. The most commonly studied type[2] is where the bin is a square, cube, or d-dimensional hypercube (for $d\geq 4$) while the items are rectangles, cuboids, and d-dimensional hyper-rectangles, respectively. More precisely, in *the 2-dimensional geometric bin packing problem* we are given a set of rectangular items I with each item having a height $h_i\in(0,1]$, width $w_i\in(0,1]$, the aim is to find the minimum cardinality partition of I into subsets $\{B_i\}$ and map each item j to a rectangular region $R_j=(l_j,l_j+w_j)\times(b_j,b_j+h_j)$, or $R_j=(l_j,l_j+h_j)\times(b_j,b_j+w_j)$ if rotations are allowed, such that (i) for any item $j\in I$, $R_j\subseteq[0,1]\times[0,1]$, and (ii) for any two items $j,j'\in B_i$, $R_j\cap R_{j'}=\emptyset$.

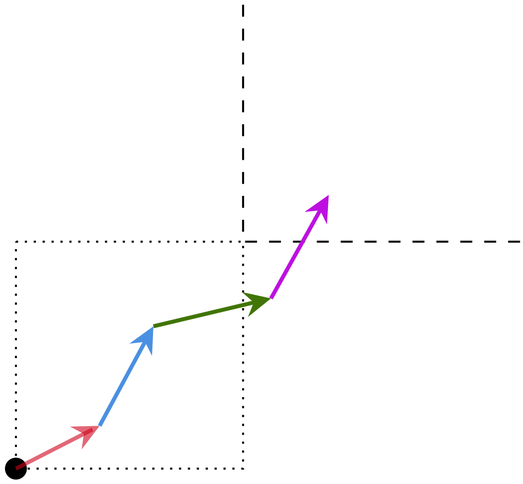
We have much stronger inapproximability results for geometric bin packing, i.e., we know that the (absolute) approximation ratio of any algorithm for the 2-dimensional geometric bin packing can not be better than 2 unless P=NP. Furthermore, it is not even possible to obtain an algorithm with asymptotic ratio better than $1+\frac{1}{3792}$ and $1+\frac{1}{2196}$ for the version with and without rotations, respectively [4, 12]. Caprara, Lodi, and Monaci [9] gave an APTAS for 2-dimensional GBP restricted to shelf packing while Caprara [8] showed that the ratio of optimal solution of 2-dimensional GBP to 2-dimensional GBP restricted to shelf packing is $T_\infty\approx 1.691$ ultimately obtaining a $T_\infty+\epsilon$ approximation. Bansal et al. [5] introduced the Round and

[1] $\gamma\approx 0.57721$ is the Euler-Mascheroni constant

[2] geometric bin packing without any qualification will refer to this type.



(a) Invalid Configuration



(b) Valid Configuration

Figure 1.3: configurations of vector bin covering

Approx framework which when used along with the shelf packing due to Caprara et al. gives an $1 + \ln(T_\infty) \approx 1.52$ approximation. Jansen and Prädel [28] improved it to 1.5 approximation using a non-trivial structure of packing. Bansal and Khan [3] used the Round and Approx framework to obtain a $\ln 1.5 + 1 \approx 1.406$ approximation. This is currently the best known algorithm for both with and without rotations.

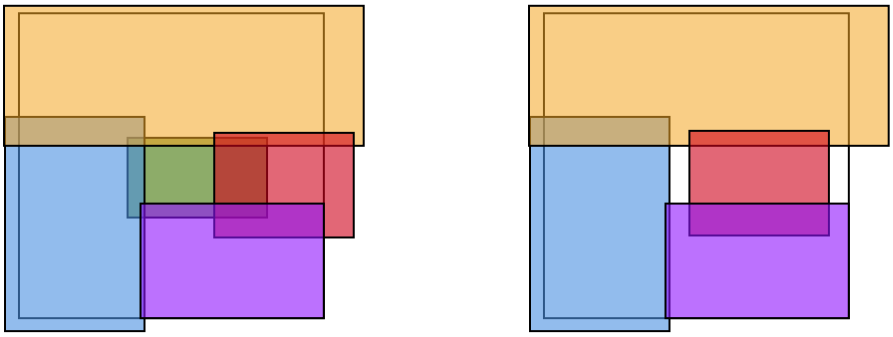
1.2.2 Bin Covering

Assmann et al. [2] were the first to look at the bin covering problem. They showed an approximation ratio of 2 for a greedy algorithm. A reduction from the partition problem can be used to show that it is NP-hard to obtain a $2-\epsilon$ absolute approximation ratio for bin covering problem. Therefore, as is the case with bin packing and its variants the asymptotic approximation ratio is the focus of current research and to that end the adjective asymptotic will be dropped hereon.

Csirik et al. [17] gave 4/3 approximation for the bin covering problem. Csirik, Johnson and Kenyon [18] obtained an APTAS for the problem. Later, Jansen and Solis-Oba [29] obtained an AFPTAS.

Similar to bin packing we can generalize it to (i) vector bin covering and (ii) geometric bin covering in higher dimensions.

In the vector bin covering problem, given a set of vectors S from $(0, 1]^d$, the aim is to obtain a disjoint family of subsets (also called bins) with the maximum cardinality such that for each bin B, we have $\sum_{v \in B} v \geq \mathbf{1}$. For vector bin covering, Alon et al. [1] gave an $O(\ln d)$ approximation for d-dimensional vector bin covering. This the best approximation algorithm



(a) valid configuration (b) invalid configuration

Figure 1.4: configurations of geometric bin covering

known for both the fixed and arbitrary dimension regimes. For the special case of $d = 2$, the same paper give a 2-approximation. In terms of inapproximability results Sandeep [41] showed a $\Omega(\frac{\log d}{\log \log d})$ lower bound for arbitrary dimensions.

For the geometric bin covering problem, it is known that there is no APTAS even in two dimensions (for both with and without rotation) [12]. There is no further literature known to the author on the geometric bin covering problem.

1.2.3 Knapsack

As we have already remarked the classical knapsack has an FPTAS (see [27, 37]).

Similar to problems discussed above knapsack can also be generalized to (i) vector knapsack (aka packing integer program) and (ii) geometric knapsack. The *m-dimensional vector knapsack*, also known as *Packing Integer Program (PIP)*, is

$$\max \sum_{j \in [m]} c_j x_j \tag{1.1a}$$

Subject to,

$$a_i^T x \leq 1 \qquad i \in [m] \tag{1.1b}$$

$$x_j \in \{0, 1\} \qquad j \in [n] \tag{1.1c}$$

where $c_j > 0$, $a_i \geq 0$. Frieze and Clarke [21] considered the case where m is a fixed constant, i.e., not part of the input and obtained a PTAS using a simple LP rounding scheme. Kulik and Sachnai [35] showed that PIP does not have an EPTAS by reducing a parameterized version of the subset sum problem, known as the sized subset sum.

For the sake of brevity we discuss only the 2-dimensional geometric knapsack (2DGK) problem. In *the 2-dimensional geometric knapsack* given a set of items I with each item i

having a height $h_i \in (0,1]$, width $w_i \in (0,1]$, and profit $p_i > 0$, the aim is obtain subset $I' \subseteq I$ and map each item in $i \in I'$ to a rectangular region $R_i = (l_i, l_i + w_i) \times (b_i, b_i + h_i)$ (or $R_i = (l_i, l_i + h_i) \times (b_i, b_i + w_i)$ if rotations are allowed) such that $\sum_{i \in I'} p_i$ is maximized while satisfying the constraint: (i) $R_i \cap R_{i'} = \emptyset$ for any $i, i' \in I'$ and (ii) $R_i \subseteq [0,1] \times [0,1]$. We get the *cardinality 2DGK* if $p_i = 1$ for all items. Leung et al. [38] showed that there is no FPTAS for 2DGK unless P=NP even for packing square (therefore even if rotations are allowed). Grandoni et al. [24] showed that there is no EPTAS for 2DGK even if rotations are allowed unless W[1]=FPT. Jansen and Zhang [31, 30] gave $(2+\epsilon)$ approximation for 2DGK even for both with and without rotations. Gálvez et al. [22] achieved an approximation ratio less than 2 for all four cases, i.e., 1.72, 1.89, $\frac{3}{2}+\epsilon$, and $\frac{4}{3}+\epsilon$ for cardinality 2DGK, weighted 2DGK, weighted 2DGK with rotation, and cardinality 2DGK with rotation, respectively by obtaining a PTAS for a problem called L-packing. It is an open problem to determine if there is PTAS for 2DGK for any of these cases. Finally we would like to remark that 3DGK is known to not have a PTAS unless P = NP [12].

In the interest of brevity we end our survey here but an interested reader can look at the survey on packing problems by Christensen et al. [14].

1.3 Organization of the thesis

The Chapter 2 discusses elementary concepts on approximation and inaproximability. Chapter 3 shows that there is no APTAS for 2-dimensional vector bin packing and 2-dimensional vector covering problem while pointing out an oversight in a previously known proof the same fact. In Chapter 4 we look at the multidimensional variants of the min-knapsack problem (covering analogue of the knapsack problem). Finally, in Chapter 5 we discuss the findings of this thesis and look at some worthwhile directions for future research on packing and covering problems.

Chapter 2

Preliminaries

2.1 Approximation Algorithms

The class of NP-hard problems has escaped attempts by many to provide an "efficient" solution. Though no proof, as of writing, is known for P$\neq$NP, it is now widely accepted among computer scientists. But, for many of these problems efficient heuristics were known. It so turns out that many of these heuristics have a bounded error with respect to the optimal solution. More precisely,

Definition 2.1. *If an algorithm A outputs a feasible solution for any instance of a given maximization (or minimization) problem, then the (absolute) approximation ratio of A is* $\sup_I \frac{OPT(I)}{A(I)}$ *(or* $\sup_I \frac{A(I)}{OPT(I)}$ *respectively).*

For some problems (e.g., bin packing) looking at the absolute approximation ratio is counter-productive as many of these problems admit algorithms which perform well in all but some edge cases and these edge cases generally consist of instances of low value solutions. To get an accurate reflection of their performance the following performance measure is generally considered:

Definition 2.2. *If an algorithm A outputs a feasible solution for any instance of a given maximization (or minimization) problem, then the asymptotic approximation ratio of A is* $\limsup_{n\to\infty} \max\left\{\frac{OPT(I)}{A(I)} | OPT(I) = n\right\}$ *(or* $\limsup_{n\to\infty} \max\left\{\frac{A(I)}{OPT(I)} | OPT(I) = n\right\}$*)*

For any given problem our aim is in general to get a class of algorithms which can give us answers arbitrarily close to the optimal, i.e., we have an algorithm with the approximation ratio of our choosing.

Definition 2.3. *A family of algorithms (which run in time polynomial in the input size)* $\{A_\epsilon\}_{\epsilon>0}$ *where* A_ϵ *has an approximation ratio of* $1+\epsilon$ *is known as a polynomial time approximation scheme (PTAS). A family of algorithms* $\{A_\epsilon\}_{\epsilon>0}$ *where* A_ϵ *has an asymptotic approximation ratio of* $1+\epsilon$ *is known as an asymptotic polynomial time approximation scheme (APTAS).*

It may be occur to the reader at this stage that surely obtaining a better solution should take more time. And indeed in most cases running time does increase with decreasing ϵ. So, to distinguish between PTAS which perform well with respect to time following notions are studied in the literature.

Definition 2.4. *A family of algorithms* $\{A_\epsilon\}_{\epsilon>0}$ *which run in time* $f(\epsilon)\cdot p(n)$ *where* n *is the input size,* f *is an arbitrary function and* p *is a polynomial with each* A_ϵ *having an approximation ratio of* $1+\epsilon$ *is known as an efficient polynomial time approximation scheme (EPTAS). A family of algorithms* $\{A_\epsilon\}_{\epsilon>0}$ *which run in time* $p(1/\epsilon, n)$ *where* n *is the input size and* p *is a polynomial with each* A_ϵ *having an approximation ratio of* $1+\epsilon$ *is known as a fully polynomial time approximation scheme (FPTAS).*

For an introduction to the topic of approximation algorithm the reader can refer the books by Williamson and Shmoys [43], and Vazirani [42].

2.2 Inapproximability

Though all the inapproximability results are based on the assumption that P $\neq$ NP or similar widely believed complexity theoretic assumptions, the techniques used to obtain such results can be quite varied even on the high level. Some of the simplest results are obtained by simply reducing an NP-hard problem A to α approximation of the problem B. These reductions rely on creating a gap between the optimal value of instances generated for a "no" instance of problem A and a "yes" instance of problem A. This means for minimization problems we show it is NP-hard to distinguish between instances for which all solution are at least some value s and instance for which there is a solution of value less than c. For example, in the reduction from partition to bin packing, we show $s = 3$ and $c = 2$. Then there are the gap preserving reductions which starting from such a gap result show another gap result. Håstad's [25] inapproximability results on Vertex Cover, Max-E3-Lin-2, Max-E3-SAT etc. are all examples of such reductions. The third family of methods relies on the study of what are called *approximation preserving reduction.*

2.2.1 Approximation preserving reductions

In study of approximation algorithm one of the most important class of problems are those for which one can find a constant approximation (whether such an approximation is known is a different question entirely). So, we define the class APX to be the class of problems with a polynomial time algorithm with constant approximation ratio. The class PTAS is the class of problems with a PTAS. A *reduction* from a problem A to B in this context will refer to a tuple of polynomial time computable functions (f, g) where if x is instance of A then $f(x)$ is an instance of B and if y is solution to $f(x)$ then $g(x, y)$ is a solution to problem A. We call a reduction *approximation preserving* if it preserves membership in APX, or PTAS, or both. Before describing such reductions we need the following definition,

Definition 2.5. *Given a solution S to an instance I of a problem P, its error $E(I, S)$ is,*

$$E(I, S) = \max \left\{ \frac{V(I, S)}{OPT(I)}, \frac{OPT(I)}{V(I, S)} \right\} - 1 \tag{2.1}$$

The following notion of reductions are known to preserve membership in APX, PTAS, respectively [16].

Definition 2.6. *Let F, G be two optimization problems, F is said to be A-reducible to G, in symbols $F \leq_A G$, if there is a reduction (f, g) from F to G such that, for any solution y of $f(x)$:*

$$E(f(x), y) \leq \epsilon \Rightarrow E(x, g(x, y)) \leq c(\epsilon) \tag{2.2}$$

where c is some function.

Definition 2.7. *Let F, G be two optimization problems, F is said to be P-reducible to G, in symbols $F \leq_P G$, if there is a reduction (f, g) from F to G such that, for any solution y of $f(x)$:*

$$E(f(x), y) \leq c(\epsilon) \Rightarrow E(x, g(x, y)) \leq \epsilon \tag{2.3}$$

where c is some function.

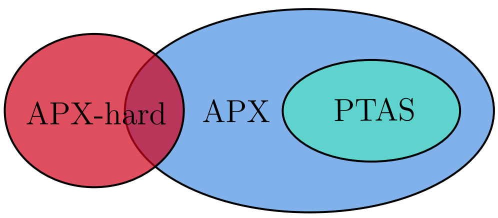


Figure 2.1: complexity classes for approximation algorithms

Finally, we define APX-hard as the class of problems P such that any problem $P' \in$ APX is P-reducible to P. Since we know there are problems in APX and not in PTAS, e.g., bin packing, and P-reductions preserve membership in PTAS, therefore, there is no PTAS possible for APX-hard problems. For a more complete treatment of this subject the reader can consult the survey by Crescenzi [15].

2.2.2 Parameterized Complexity and Non-Existence of EPTAS

Till now we have considered only one possible solution to dilemma presented to us by NP-hard problems, i.e., approximation algorithms which try to find a close to optimal solution to all instances of the problem under consideration. The other direction to consider is to look at obtaining an exact solution to a subset of instances. In this section we turn our attention to decision problems. An instance of a *parameterized problem* is of form (x, k) where k is called the *parameter*. For this discussion assume that k is a positive integer. We can *slice* a given parametrized problem L into $L_k = \{(x, k)|(x, k) \in L\}$ for each k. L_k is called the k-th slice of L. The objective here is to find algorithms whose runtime is a polynomial for any fixed slice. That is we want algorithms whose runtime is at most $f(k)|x|^c$ where c is a constant.

Definition 2.8. *We say a (parameterized) problem P is fixed-parameter tractable (FPT) if there is an algorithm A, a constant c and function f such that A can decide if $(x, k) \in P$ in a most $f(k)|x|^c$ steps.*

In context of parameterized algorithms the appropriate notion of reduction would be one which retains membership in FPT.

Definition 2.9. *We say a (parameterized) problem P fixed parameterized reducible to P' if there are functions f, g, h, and a constant c such that (i) $(f(x, k), g(k)) \in L'$ iff $(x, k) \in L$ and (ii) $f(x, k)$ is computable in time $h(k)|x|^c$.*

There is a class of problems called W[1] for which it is believed that W[1] $\neq$ FPT. Hence, showing a problem is W[1]-hard is sufficient to show it is not in FPT.

Finally, for optimization problems the *natural parameterization* is to have x as the original instance and k as the value of the optimal solution. If for such a natural parameterization we are able to show that it is not in FPT then there is no EPTAS for the original optimization problem. In fact, if there was an EPTAS for the natural parameterization of such a problem then it is easy to see $A_\epsilon(x) \geq k$ if and only if OPT $\geq k$ where $\epsilon = 1/2k$ (for a minimization problem).

Chapter 3

Vector Bin Packing

3.1 Overview

It was believed that Woeginger [44] showed that there is no APTAS for vector bin packing with $d \geq 2$. However, as we show in Section 3.3, there was a minor oversight in the original proof. Specifically, an essential claim made in the proof fails to hold for a few special cases. We also examine some natural modifications to the claim, which exclude those special cases. We conclude that it is impossible to obtain the final result if we try to use the same arguments. Unfortunately, this oversight is also present in the $\frac{391}{390}$ lower bound for vector bin packing by Chlebík and Chlebiková [13]. Hence, we present a revision to the original proof in Section 3.2. Although our proof uses essentially the same construction as the original proof, the final analysis is slightly different, and the main ideas for the analysis are borrowed from [4, 12]. Specifically, we have obtained a gap reduction instead of an approximation ratio preserving reduction. The APX-hardness of vector bin packing, though not stated explicitly, was considered to be a simple corollary of the original result (see [4]) which showed an approximation preserving reduction. Although the revised proof gives us a constant lower bound on the approximation ratio, we cannot conclude that the vector bin packing problem is APX-hard. Though Sandeep [41] showed that the best approximation factor any algorithm can have, for high enough value of d, is $\Omega(\log d)$, We note that Sandeep's lower bound of $\Omega(\ln d)$ does not hold for low dimensions, and hence, it does not even rule out the possibility of APTAS in the 2-dimensional case. Finally, we adapt our proof for vector bin covering to show that there is no APTAS for vector bin covering with dimension $d \geq 2$.

As is the case in [44] and [4], we start with maximum 3-dimensional matching (denoted by MAX-3-DM) and reduce it to an instance of 4-Partition and finally reduce it to a vector bin packing instance. In a 3-dimensional matching instance we have three sets $X = \{x_1, x_2, \ldots, x_q\}$,$Y =$

$\{y_1, y_2, \ldots, y_q\}$, and $Z = \{z_1, z_2, \ldots, z_q\}$ and we are given a set of tuples $T \subseteq X \times Y \times Z$. The aim is to find the maximum cardinality subset $T' \subseteq T$ such that no element from X, Y, or Z occurs in more than one tuple. In the bounded variant of MAX-3-DM (denoted by MAX-3-DM-B), it is assured that any element which belongs to either X, Y, or Z will appear in at most B tuples. Kann [32] showed that bounded maximum matching with bound 3 is MAX SNP-hard which in turn implies it is APX-hard. Later, it was shown by Petrank [39] that it is NP-hard to distinguish between instances where there is a solution T' with $|T'| = q$ and from instances for which every solution T', $|T'| \leq (1-\epsilon)q$ for a constant ϵ. There is also a more restricted variant of the problem, which is frequently studied where there are exactly B tuples for each element of the sets X,Y, and Z called the exact maximum 3-dimensional matching (denoted by MAX-3-DM-EB). In case of MAX-3-DM-E2 it was shown by Berman and Karpinski [7] that it is NP-hard to approximate with ratio better than $\frac{98}{97}$, which Chlebík and Chlebíková [11] improved to $\frac{95}{94}$. Finally, Chlebík and Chlebíková [12] also note an useful corollary of their $\frac{95}{94}$ bound, which is the following result for the promise variant of MAX-3-DM-E2,

Theorem 3.1. *[12] Let I_M be a MAX-3-DM-E2 instance comprising of sets X, Y, Z and tuples $T \subseteq X \times Y \times Z$ with $|X| = |Y| = |Z| = q$. Then, it is NP-hard to distinguish between the case with $OPT(I_M) \geq \lceil \beta_0 q \rceil$ and $OPT(I_M) \leq \lfloor \alpha_0 q \rfloor$, where $\alpha_0 = 0.9690082645$ and $\beta_0 = 0.979338843$.*

Finally, for our vector bin covering result we use the same reduction to 4-Partition to finally obtain a reduction to a vector bin covering instance.

3.2 Vector Bin Packing has no APTAS

In this section, we prove our main result, i.e., there is no APTAS for vector bin packing. We do so by modifying the construction in the original proof given in [44] by adding a set of dummy vectors[1]. The final analysis is based on the analysis in [4] for the geometric bin packing lower bound.

We start by defining a few integers based on the given MAX-3-DM instance I_M. Let $r = 64q$, where $q = |X| = |Y| = |Z|$ and $b = r^4 + 15$. Define integers x'_i, y'_i, z'_i corresponding to

[1]We have tried to stay as close to the original proof as possible in terms of notations and arbitrary choices. Yet we have made two notable changes, (i) using $r = 64q$, and (ii) using $t_{(i,j,k)}$ to denote a tuple.

$x_i \in X, y_i \in Y, z_i \in Z$ to be,

$$\begin{aligned} x_i' &= ir + 1, \\ y_i' &= ir^2 + 2, \\ z_i' &= ir^3 + 4, \end{aligned}$$

and for $t_{(i,j,k)} = (x_i, y_j, z_k) \in T$ define $t'_{(i,j,k)}$ as,

$$t'_{(i,j,k)} = r^4 - kr^3 - jr^2 - ir + 8.$$

Let U' be the set of integers constructed as above. Also, note that for any integer $a' \in U'$ constructed as above we have $0 < a' < b$. These integers were constructed so that the following statement holds (cf. Observation 2 in [44]).

Lemma 3.1. *A set of four integers from U' add up to b if and only if they correspond to some elements $x_i \in X, y_j \in Y, z_k \in Z$ and tuple $t_{(i,j,k)} \in T$ where $t_{(i,j,k)} = (x_i, y_j, z_k)$.*

Proof. (If) Suppose $x_i \in X, y_j \in Y, z_k \in Z$ and $t_{(i,j,k)} \in T$ where $t_{(i,j,k)} = (x_i, y_j, z_k)$ then it is easy to verify that indeed

$$t'_{(i,j,k)} + x_i' + y_j' + z_k' = b$$

(Only if) Conversely, suppose that four integers a_1', a_2', a_3', a_4' sum to b. Considering the equation modulo r and using the fact $1 + 2 + 4 + 8$ is the only possible way of obtaining 15 as a sum of four elements (possibly with repetition) from the set $\{1, 2, 4, 8\}$ and therefore we conclude the integers must correspond one element each from X, Y, Z, T. This means we can write a_1', a_2', a_3', a_4' as $x_i', y_j', z_k', t'_{(i_0,j_0,k_0)}$. Now, considering the equation modulo r^2, r^3, r^4 gives us $i = i_0$,$j = j_0$, and $k = k_0$. □

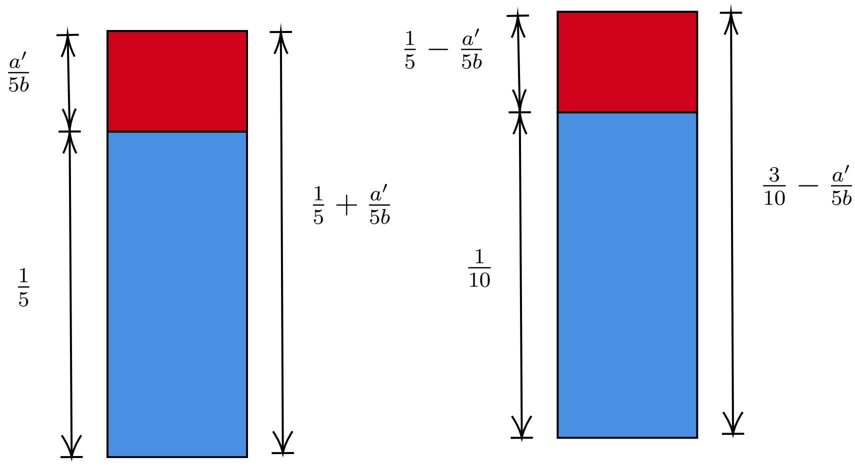
To obtain a vector bin packing instance for each integer a' constructed above construct the following vector,

$$\mathbf{a} = \left(\frac{1}{5} + \frac{a'}{5b}, \frac{3}{10} - \frac{a'}{5b} \right).$$

We also construct additional $|T| + 3q - 4\beta(I_M)$ dummy vectors as follows,

$$\mathbf{d} = \left(\frac{3}{5}, \frac{3}{5} \right),$$

where $\beta(\cdot)$ is an arbitrary function which will be fixed later. We now note a few properties of



(a) first component

(b) second component

Figure 3.1: non-dummy vectors

the vectors. First of these pertains to how many vectors can fit in a bin (cf. Observation 4, Lemma 2.5 from [44] and [4]).

Lemma 3.2. *A bin can contain at most* 4 *vectors. If a bin contains a dummy vector it can contain at most one more vector. Furthermore, two dummy vectors will not fit in a bin while any other set of two vectors fit in a bin.*

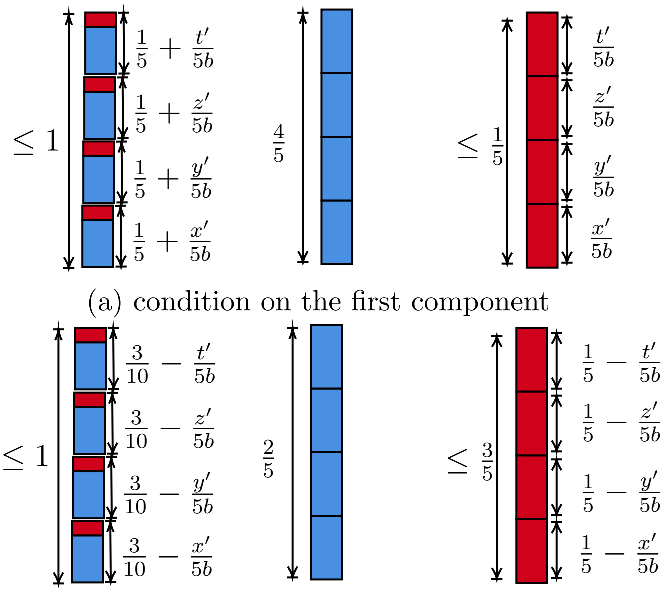
Proof. The first part follows from the fact that the first component of any vector is strictly greater than $\frac{1}{5}$. The second part of the claim follows from the fact any vector in the instance has first component greater than $\frac{1}{5}$ and the dummy vector has first component equal to $\frac{3}{5}$. For the third part, observe if there are two dummy vectors then both the components would add up to $2 \cdot \frac{3}{5} > 1$. However, if one of them is a non-dummy vector, then notice that both components of a non-dummy vector are less than $\frac{2}{5}$, and both components of any vector are less than $\frac{3}{5}$. Hence, both components of the sum are less than 1. $\square$

The following lemma shows that a configuration corresponding to a tuple is an optimal configuration (cf. Observation 3 from [44]).

Lemma 3.3. *A set of four vectors fits in a bin if and only if it corresponds to a tuple.*

Proof. (If) For a tuple $t_{(i,j,k)} = (x_i, y_j, z_k)$, we have $t'_{(i,j,k)} + x'_i + y'_j + z'_k = b$ by Lemma 3.1. So, we have

$$\mathbf{t}_{(i,j,k)} + \mathbf{x}_i + \mathbf{y}_j + \mathbf{z}_k = \left(\frac{4}{5} + \frac{t'_{(i,j,k)} + x'_i + y'_j + z'_k}{5b}, \frac{6}{5} - \frac{t'_{(i,j,k)} + x'_i + y'_j + z'_k}{5b} \right) = (1, 1).$$



(b) condition on the second component

Figure 3.2: The only if direction of Lemma 3.3

(Only if) Suppose there are four vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ which fit in a bin. By Lemma 3.2 all the vectors are non-dummy vectors. Hence, each vector can be written as:

$$\mathbf{a}_i = \left(\frac{1}{5} + \frac{a'_i}{5b}, \frac{3}{10} - \frac{a'_i}{5b}\right).$$

As they fit in a bin we get the following two conditions,

$$\frac{4}{5} + \frac{\sum_{i=1}^{4} a'_i}{5b} \leq 1,$$
$$\frac{6}{5} - \frac{\sum_{i=1}^{4} a'_i}{5b} \leq 1,$$

which simplify to $\sum_{i=1}^{4} a'_i \leq b$ and $\sum_{i=1}^{4} a'_i \geq b$. Combining the inequalities we get $\sum_{i=1}^{4} a'_i = b$. Therefore, by Lemma 3.1 the vectors correspond to a tuple. □

Now, we show that the above construction is a gap reduction from MAX-3-DM to 2-dimensional vector packing (cf. Theorem 2.1 from [4]),

Lemma 3.4. *If a MAX-3-DM instance I_M has a solution with $\beta(I_M)$ tuples the constructed vector bin packing instance has a solution with $|T| + 3q - 3\beta(I_M)$ bins. Otherwise, if all the solutions of the MAX-3-DM instance have at most $\alpha(I_M)$ tuples then the constructed instance needs at least $|T| + 3q - \frac{\alpha(I_M)}{3} - \frac{8\beta(I_M)}{3}$ bins where $\alpha(\cdot)$ is an arbitrary function.*

Proof. First, we show that if a MAX-3-DM instance has a matching consisting of $\beta(I_M)$ tuples, then the vector bin packing instance has a solution of $|T|+3q-3\beta(I_M)$ bins. Using Lemma 3.3, the $4\beta(I_M)$ vectors corresponding to the $\beta(I_M)$ tuples and their elements can be packed into $\beta(I_M)$ bins. Each of the $|T| + 3q - 4\beta(I_M)$ non-dummy vectors can be packed along with a dummy vector into $|T| + 3q - 4\beta(I_M)$ bins by Lemma 3.2.

Now, suppose that for a given instance, all the solutions have at most $\alpha(I_M)$ tuples. Let n_g be the number of bins with 4 vectors, n_d be the number of bins with dummy vectors, and n_r be the rest of the bins. Now, since any solution to the bin packing instance must cover all the non-dummy vectors,

(a) any bin containing four vectors consists of only non-dummy vectors by Lemma 3.3,

(b) any bin containing a dummy vector contains at most one non-dummy vector, by Lemma 3.2, and

(c) any other bin can contain at most 3 vectors by Lemma 3.2.

Therefore, we have

$$4n_g + 3n_r + n_d \geq 3q + |T|.$$

Now, by Lemma 3.2 we have $n_d = |T| + 3q - 4\beta(I_M)$. Hence, the above inequality simplifies to,

$$\begin{aligned} &4n_g + 3n_r \geq 4\beta(I_M) \\ &\Rightarrow n_g + n_r \geq \frac{4}{3}\beta(I_M) - \frac{n_g}{3} \\ \Rightarrow n_g + n_r + n_d \geq |T| + 3q - \frac{n_g}{3} - \frac{8}{3}\beta(I_M) \qquad &[\text{Since, } n_d = |T| + 3q - 4\beta(I_M)]. \end{aligned}$$

Since there are at most $\alpha(I_M)$ triples in the MAX-3-DM instance, by Lemma 3.3 we have $n_g \leq \alpha(I_M)$. Therefore, the number of bins is at least

$$|T| + 3q - \frac{\alpha(I_M)}{3} - \frac{8\beta(I_M)}{3}.$$

□

At this point the reader may want to step back and look at the justification for the seeming arbitrary choice for the number of dummy vectors, i.e., $|T|+3q-4\beta(I_M)$. In fact, if they carefully examine the proof of Lemma 3.4 they may find that it is precisely the number of dummy vectors which increases the "soundness" value of the gap without increasing the "completeness". Also, note that without the use of dummy vectors the gap obtained would be trivial. The following inaproximability for vector bin packing directly follows from Lemma 3.4.

Theorem 3.2. *There is no APTAS for the d-dimensional vector bin packing with $d \geq 2$ unless P=NP. Furthermore, for the 2-dimensional case there is no algorithm with asymptotic approximation ratio better than $\frac{600}{599}$.*

Proof. Suppose that there is an algorithm with approximation ratio $1 + \frac{\beta_0 - \alpha_0}{15 - 9\beta_0}$. Then we can distinguish between MAX-3-DM-E2 instances (i) having a solution of $\lceil \beta_0 q \rceil$ triples and (ii) having no solutions with more than $\lfloor \alpha_0 q \rfloor$ tuples using Lemma 3.4 with $\alpha(I_M) = \lfloor \alpha_0 q \rfloor$ and $\beta(I_M) = \lceil \beta_0 q \rceil$. By Theorem 3.1, we know it is NP-hard to distinguish between these two types of MAX-3-DM-E2 instances with $\beta_0 = 0.979338843$, and $\alpha_0 = 0.9690082645$. Hence, we obtain the bound of $1 + \frac{\beta_0 - \alpha_0}{15 - 9\beta_0}$. Simple calculations will show this is at least $1 + \frac{1}{599}$. □

Note that the above proof does not show that the vector bin packing problem is APX-hard.

3.3 Woeginger's Proof

In this section we look at the original proof of non-existence of APTAS for vector bin packing by Woeginger while showing that it has a minor error. We show that using a counterexample.

Woeginger's proof uses essentially the same reduction as ours, i.e., there we had $r = 32q$, $b = r^4 + 15$ and then for each $x_i \in X, y_i \in Y, z_i \in Z$ we had,

$$x'_i = ir + 1,$$
$$y'_i = ir^2 + 2,$$
$$z'_i = ir^3 + 4,$$

and for $t_l \in T$ was t'_l defined by,

$$t'_l = r^4 - kr^3 - jr^2 - ir + 8.$$

And finally, to obtain a vector bin packing instance for each integer a' constructed above construct the following vector,

$$\mathbf{a} = \left(\frac{1}{5} + \frac{a'}{5b}, \frac{3}{10} - \frac{a'}{5b}\right).$$

The above set of vectors forms a 2-dimensional vector bin packing instance $\mathbf{U}$. A noticeable difference from our reduction being the absence of dummy vectors.[1] In [44], Woeginger claimed that,

Claim 3.1 (Observation 4 in [44]). *Any set of 3 vectors in* $\mathbf{U}$ *can be packed in a unit-bin. No set of 5 vectors in* $\mathbf{U}$ *can be packed into a unit-bin.*

We show that this claim does not hold in general. In particular, all sets of 3 vectors can not be packed into a unit-bin. The idea is that the integers corresponding to tuples are quite large and consequently the first component of these vectors are quite large. Hence, a set of 3 vectors can not fit in a bin.

Consider the tuple vectors for the tuples $t_1 = (x_1, y_1, z_1), t_2 = (x_2, y_1, z_1)$, and $t_3 = (x_3, y_1, z_1)$, i.e., $\mathbf{t}_1, \mathbf{t}_2$, and $\mathbf{t}_3$, respectively. According to the claim, the vectors $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ corresponding to the above tuples can be packed in a bin. Suppose $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ can indeed be packed in a bin. This implies that the first components of the vectors do not exceed 1, i.e.,

$$\frac{3}{5} + \frac{t_1' + t_2' + t_3}{5b} \leq 1,$$

which simplifies to,

$$t_1' + t_2' + t_3' \leq 2b.$$

Finally, using

$$\begin{aligned} t_1' &= r^4 - r^3 - r^2 - r + 8, \\ t_2' &= r^4 - r^3 - r^2 - 2r + 8, \\ t_3' &= r^4 - r^3 - r^2 - 3r + 8, \end{aligned}$$

and

$$b = r^4 + 15,$$

along with further simplification we get,

$$r^4 \leq 3r^3 + 3r^2 + 6r + 6.$$

[1] Also notice that $r = 32q$ and tuples are denoted by t_l.

But this inequality does not even hold for $r \geq 32$ whereas 32 is the smallest value for $r = 32q$. Thus the claim is incorrect. This implies his main lemma (Lemma 5 in [44]) fails to hold.

Claim 3.2 (Lemma 5 in [44])**.** *Let $\alpha > 0$ be an integer such that $|T| - \alpha$ is divisible by 3. Then there exists a feasible solution for the instance I_M for MAX-3-DM that contains at least α triples if and only if there exists a feasible packing for the instance $\mathbf{U}$ of the 2-dimensional vector packing problem that uses at most $|T| + \frac{1}{3}(q - \alpha)$ unit bins.*

We now consider, and rule out, a few natural attempts at fixing the proof. As a first attempt, we modify the claim of Observation 4 from [44] to exclude the case of 3 tuples. In this case, we see that it is not possible prove the original claim of Lemma 5 from [44]. Another attempt inspired by the failures of the above attempt is to consider modifying the claim of Observation 4 to apply to any set of 2 vectors. In this case, it is clear that the original claim of Lemma 5 can not be proven and even the natural modification to the claim of Lemma 5 to consider at most $\frac{1}{2}(3q + |T|) - \alpha$ bins instead of $q + \frac{1}{3}(q - \alpha)$ bins can not be proven. It seems that it may not be possible to show that the construction given in [44] is an approximation preserving reduction. Finally, the analysis in [13] also uses the relation in Lemma 5 of Woeginger's proof to obtain their $\frac{391}{390}$ inapproximability which unfortunately means that their analysis also suffers from the same oversight. Hence, the need for the revision.

3.4 Vector Bin Covering has no APTAS

In this section, we prove that vector bin covering has no APTAS unless P=NP by adapting the proof presented in Section 3.2. The analysis is slightly more complicated and bears some resemblance to the analysis of the reduction to geometric bin covering problem presented in [12]. To show this we obtain a gap preserving reduction from MAX-3-DM to 2-dimensional vector bin covering. We start with the same set of integers U' we had in Section 3.2. To obtain a vector bin covering instance for each integer a' in U' construct the following vector,

$$\mathbf{a} = \left(\frac{1}{5} + \frac{a'}{5b}, \frac{3}{10} - \frac{a'}{5b}\right).$$

We also construct additional $|T| + 3q - 4\beta(I_M)$ dummy vectors as follows,

$$\mathbf{d} = \left(\frac{4}{5}, \frac{4}{5}\right),$$

where $\beta(\cdot)$ is an arbitrary function which will be fixed later.[1] If a unit cover (or bin) has at least one dummy vector then we call it a D-bin. Otherwise, if a unit cover has no dummy vectors the we call it a non-D-bin.

Observation 3.1. *Any set of 5 vectors can cover a bin. Any vector along with a dummy vector can cover a bin. At least 2 vectors are needed to form a unit cover.*

Though, as we observe, any bin can be covered using a dummy vector along with another vector. We now show that for a non-D-bin the optimal cover has cardinality 4.

Lemma 3.5. *A set of four vectors covers a non-D-bin if and only if it corresponds to a tuple.*

Proof. (If) For a tuple $t_{(i,j,k)} = (x_i, y_j, z_k)$, we have $t'_{(i,j,k)} + x'_i + y'_j + z'_k = b$ by Lemma 3.1. So, we have

$$\mathbf{t}_{(i,j,k)} + \mathbf{x}_i + \mathbf{y}_j + \mathbf{z}_k = \left(\frac{4}{5} + \frac{t'_{(i,j,k)} + x'_i + y'_j + z'_k}{5b}, \frac{6}{5} - \frac{t'_{(i,j,k)} + x'_i + y'_j + z'_k}{5b}\right) = (1,1).$$

(Only if) Suppose there are four vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ which cover a non-D-bin. By our assumption all the vectors are non-dummy vectors. Hence each vector can be written as,

$$\mathbf{a}_i = \left(\frac{1}{5} + \frac{a'_i}{5b}, \frac{3}{10} - \frac{a'_i}{5b}\right).$$

As they cover a bin we get the following two conditions,

$$\frac{4}{5} + \frac{\sum_{i=1}^{4} a'_i}{5b} \geq 1,$$
$$\frac{6}{5} - \frac{\sum_{i=1}^{4} a'_i}{5b} \geq 1,$$

which simplify to $\sum_{i=1}^{4} a'_i \geq b$ and $\sum_{i=1}^{4} a'_i \leq b$. Combining the inequalities, we get $\sum_{i=1}^{4} a'_i = b$. Therefore, by Lemma 3.1 the vectors correspond to a tuple. □

Now, we are ready to prove our main lemma showing our reduction is indeed a gap preserving reduction.

[1] Note the difference in size of dummy vectors from Section 3.2.

Lemma 3.6. *If a MAX-3-DM instance I_M has a solution with $\beta(I_M)$ tuples then there is a solution to the vector bin covering instance with $|T| + 3q - 3\beta(I_M)$ tuples. Otherwise, if all the solutions of I_M have at most $\alpha(I_M)$ tuples then the constructed instance can cover at most $|T| + 3q - \frac{16}{5}\beta(I_M) + \frac{\alpha(I_M)}{5}$ bins, where $\alpha(\cdot)$ is an arbitrary function.*

Proof. Suppose that $\mathrm{OPT}(I_M) \geq \beta(I_M)$. Then we can cover $\beta(I_M)$ bins using $4\beta(I_M)$ vectors corresponding to the $\beta(I_M)$ tuples from the solution of I_M using Lemma 3.5. Now, we have $|T|+3q-4\beta(I_M)$ non-dummy vectors left along with exactly $|T|+3q-4\beta(I_M)$ dummy vectors. These vectors, by Observation 3.1, can cover $|T| + 3q - 4\beta(I_M)$ bins.

Now, suppose that every solution of the MAX-3-DM instance has value at most $\alpha(I_M)$. Consider an optimal solution to the constructed vector bin covering instance. We can normalize such a solution without any loss in the number of bins covered as follows,

(a) *Each bin has at most one dummy element.* Clearly, more than $|T| + 3q - 4\beta(I_M)$ bins are covered in an optimal solution. So, if there are n bins with two dummy vectors then there must be at least n non-D-bins. Therefore, we can pick one non-D-bin for each bin with 2 dummy vectors, which must contain 2 non-dummy vectors by Observation 3.1. Again by Observation 3.1, we can obtain two unit cover each with one dummy vector and one non-dummy vector. Similar arguments can be used for bins having k dummy vectors, i.e., there are at least $(k-1)n$ non-D-bins and then similar rearangements can be done to obtain kn unit covers with one dummy vector.

(b) *No subset of a unit cover is an unit cover.* To that end, some vectors can left out, i.e., they may be designated as not part of any unit cover. Now, using arguments similar to (a) we see that every dummy vector must be part of a unit cover. Also, note that number of such vectors can be at most four as five vectors always form a unit cover.

Let n_d be the number of D-bins, n_g be the number of non-D-bins covered by 4 vectors and n_r be the number of non-D-bin covered by 5 vectors. By our normalization every dummy vector is part of unit cover with exactly one dummy vector, i.e., $n_d = |T| + 3q - 4\beta(I_M)$. Since there are $3q + |T|$ non-dummy vectors,

$$\begin{aligned}
n_d + 4n_g + 5n_r &\leq 3q + |T| \\
\Rightarrow 4n_g + 5n_r &\leq 4\beta(I_M) \qquad\qquad [\text{Since, } n_d = |T| + 3q - 4\beta(I_M)] \\
\Rightarrow n_g + n_r &\leq \frac{4}{5}\beta(I_M) + \frac{n_g}{5}.
\end{aligned}$$

By Lemma 3.5, we have $n_g \le \alpha(I_M)$. Therefore,

$$\begin{aligned} n_g + n_r &\le \frac{4}{5}\beta(I_M) + \frac{\alpha(I_M)}{5} \\ \Rightarrow n_d + n_g + n_r &\le |T| + 3q - \frac{16}{5}\beta(I_M) + \frac{\alpha(I_M)}{5}. \end{aligned}$$

In other words, the number of bins covered is at most

$$|T| + 3q - \frac{16}{5}\beta(I_M) + \frac{\alpha(I_M)}{5}.$$

□

Theorem 3.3. *There is no APTAS for d-dimensional vector covering with $d \ge 2$ unless P=NP. Furthermore, for the 2-dimensional vector bin covering there is no algorithm with asymptotic approximation ratio better than $\frac{998}{997}$.*

Proof. Suppose there is an algorithm with approximation ratio $1 + \frac{\beta_0 - \alpha_0}{25 - 16\beta_0 + \alpha_0}$. Then we can distinguish between MAX-3-DM-E2 instances (i) having a solution of $\lceil \beta_0 q \rceil$ tuples and (ii) with all solution less than $\lfloor \alpha_0 q \rfloor$ using Lemma 3.6 with $\alpha(I_M) = \lfloor \alpha_0 q \rfloor$ and $\beta(I_M) = \lceil \beta_0 q \rceil$. By Theorem 3.1, we know it is NP-hard to distinguish between these two types of MAX-3-DM-E2 instances with $\beta_0 = 0.979339943$, and $\alpha_0 = 0.9690082645$. Hence, we obtain the bound of $1 + \frac{\beta_0 - \alpha_0}{25 - 16\beta_0 + \alpha_0}$. Simple calculations will show this is at least $1 + \frac{1}{997}$. □

Chapter 4

Min-Knapsack

In this chapter we study the multidimensional variants of the min-knapsack problem.[1] The min-knapsack problem is the covering analogue of the knapsack problem. In min-knapsack problem given a set of items I with sizes $s_i \in (0,1]$ and cost c_i for each item $i \in I$, the aim is to find the minimum cost cover for the knapsack, i.e., find a subset $I' \subseteq I$ which minimizes $\sum_{i \in I'} c_i$ under the constraint $\sum_{i \in I'} s_i \geq 1$.

4.1 Covering Integer Program

Recall that for the case where the dimension m is fixed considered the case where m is fixed, i.e., not part of the input Frieze and Clarke [21] obtained a PTAS using a simple LP rounding scheme. The covering variant of the above problem is the vector min-knapsack also known as the covering integer program which is,

$$\min \sum_{j \in [m]} c_j x_j \tag{4.1a}$$

Subject to,

$$a_i^T x \geq 1 \qquad i \in [m] \tag{4.1b}$$

$$x_j \in \{0,1\} \qquad j \in [n] \tag{4.1c}$$

where $c_j > 0$, $a_i \geq 0$. In the above problems, the variables x_j can be thought of as denoting exclusion or inclusion of the item j in a m-dimensional knapsack.

[1] We have been made aware that the results which appear in this chapter are already part of a paper by Kulik, Shachnai, Shmueli and Sayegh [36]. Specifically, the PTAS and the non-existence of EPTAS have appeared in their work.

4.2 Non-existence of EPTAS for CIP

Kulik and Sachnai [35] showed that PIP does not have an EPTAS by reducing a parameterized version of the subset sum problem, known as the sized subset sum. In the sized subset sum problem, given a set of positive integers $L = \{x_1, \ldots, x_n\}$, and the positive integers S, k, the aim is to decide if there is a subset $L' \subseteq L$ of size k, such that $\sum_{i \in L'} x_i = S$. The sized subset problem described by (L, S, k) is known to be W[1]-hard (see [20]).

We now give an almost identical proof of non-existence of EPTAS for CIP. First we construct a new instance $(\tilde{L}, S, k)$ where $\tilde{L} = \{\tilde{x}_1, \ldots, x_n\}$ with

$$\tilde{x}_i = \frac{x_i + \frac{k-1}{k} S}{k}$$

Note that $0 \leq \tilde{x}_i \leq \frac{2S}{k}$ if $x_i \leq S$, which we can assume without loss of generality. Simple calculations will show that the following lemma holds.

Lemma 4.1 ([35]). *The instance (L, S, k) is satisfiable if and only if $(\tilde{L}, S, k)$ is satisfiable.*

From (L, S, k) define an instance $R(L, S, k)$ of 2-dimensional CIP. For each item x_j we obtain an item j with $c_j = 1$, $a_{1,j} = \frac{\tilde{x}_j}{S}$, and $a_{2,j} = \frac{2}{k} - \frac{\tilde{x}_j}{S}$.

Lemma 4.2. *$OPT(R(L, S, k)) \geq k$.*

Proof. Assume there is a feasible subset of items $A \subseteq [n]$ whose cost is smaller than k for $R(L, S, k)$, then $|A| \leq k - 1$. Since A is feasible, we have $\sum_{j \in A} a_{1,j} = \sum_{j \in A} \frac{\tilde{x}_j}{S} \geq 1$, which means

$$1 \leq \sum_{j \in A} a_{2,j} = \sum_{j \in A} \frac{2}{k} - \frac{\tilde{x}_j}{S} = |A| \cdot \frac{2}{k} - \sum_{j \in A} \frac{\tilde{x}_j}{S} < 1,$$

a contradiction. □

Lemma 4.3. *$(\tilde{L}, S, k)$ is satisfied if and only if $OPT(L, S, k) = k$.*

Proof. Suppose the instance $(\tilde{L}, S, k)$ is satisfied then there is a subset $\tilde{L}' \subseteq \tilde{L}$ such that $\sum_{\tilde{x} \in \tilde{L}'} \tilde{x} = S$. In this case $A = \left\{ j | \tilde{x}_j \in \tilde{L}' \right\}$ is a feasible solution for $R(L, S, k)$ as $\sum_{j \in A} a_{1,j} = \frac{\sum_{j \in A} \tilde{x}_j}{S} = 1$ and $\sum_{j \in A} a_{2,j} = \frac{2|A|}{k} - \frac{\sum_{j \in A} \tilde{x}_j}{S} = 1$. Therefore by Lemma 4.2, $\mathrm{OPT}(R(L, S, k)) = k$.

Suppose $\mathrm{OPT}(R(L, S, k)) = k$ then there is a $A \subseteq [n]$ such that $\sum_{j \in A} a_{1,j} = 1$ and $\sum_{j \in A} a_{2,j} = 1$. $\sum_{j \in A} a_{1,j} = \frac{\sum_{j \in A} \tilde{x}_j}{S} = 1$ implies that $\sum_{j \in A} \tilde{x}_j = S$, which in turn means that $(\tilde{L}, S, k)$ is satisfied by $\{\tilde{x}_j | j \in A\}$. □

Corollary 4.1. *(L, S, k) is satisfiable if and only if $OPT(R(L, S, k)) = k$.*

Now we prove the main lemma needed for showing non-existence of EPTAS.

Lemma 4.4. *Let A be an approximation scheme for 2-dimensional Covering Integer Program which on input I and an error parameter ϵ runs in $f(1/\epsilon)|I|^{g(1/\epsilon)}$. Then there is an algorithm for sized subset sum with a running time of $f(2k)|(L,S,k)|^{\mathcal{O}(g(2k))} + p(|(L,S,k)|)$ where f, g are arbitrary functions and p is a polynomial.*

Proof. Consider the algorithm for sized subset sum which on input (L,S,k) generates $I = R(L,S,k)$ and runs A with error $\epsilon = 1/2k$. If $A(I) = k$ output satisfiable, otherwise output not satisfiable.

Note that if $\mathrm{OPT}(I) \geq k$ then $A(I) \geq k$ while if $\mathrm{OPT}(I) < k$ then $A(I) < k$ as $k(1+1/2k) = k + \frac{1}{2}$. Therefore, by Corollary 4.1 the algorithm under consideration decides sized subset sum correctly.

Now, the construction needs $p(|(L,S,k)|)$ time where p is a polynomial, and running A on I needs $f(2k)|R(L,S,k)|^{g(2k)}$, i.e., $f(2k)|(L,S,k)|^{(g(2k))}$. □

Hence we have,

Theorem 4.1. *There is no EPTAS for the covering integer program for $m \geq 2$ unless $W[1] = FPT$. Furthermore, the standard parameterization of covering integer program is $W[1]$-hard.*

Proof. Take g to be the constant function in Lemma 4.4. □

Here the standard parameterization of covering integer program refers to problem of deciding given an CIP instance I and a value k whether $\mathrm{OPT}(I) \leq k$.

4.3 PTAS for CIP

We show that a LP rounding scheme similar to one given by Frieze and Clarke also produces a PTAS hence verifying their remark in [21]. To that end we first consider the LP relaxation of (4.1), i.e.,

$$\min \sum_{j\in[m]} c_j x \tag{4.2a}$$

Subject to,

$$a_i^T x \geq 1 \qquad i \in [m] \tag{4.2b}$$

$$0 \leq x_j \leq 1 \qquad j \in [n] \tag{4.2c}$$

For a set $S \subseteq N = [n]$, define $T(S) = \{j \in N \setminus S | c_j > \min_{k \in S} c_k\}$. Define IP($S$) and LP($S$) to be (4.1) and (4.2), respectively, with the following additional constraints,

$$x_j = \begin{cases} 0 & j \in S \\ 1 & j \in T(S) \end{cases} \tag{4.3}$$

Finally, let $x^B(S)$ denote the basic optimal solution to LP(S).

Given $\epsilon > 0$ we set $k = \min(n, \lceil m(1+\epsilon)/\epsilon \rceil)$. To find our solution to CIP we consider the sets $S \subseteq$ with $|S| \leq k$, obtain $x^B(S)$ and then it round up to $\lceil x^B(S) \rceil = \left(\lceil x_j^B(S) \rceil\right)$. The required solution is $\hat{x} = \left\lceil x^B(\hat{S}) \right\rceil$ with the value $\hat{z} = \min_{S||S| \leq k} \sum_{j=1}^n c_j \left\lceil x_j^B(S) \right\rceil$ where $\hat{S} = \operatorname{argmin}_{S||S| \leq k} \sum_{j=1}^n c_j \left\lceil x_j^B(S) \right\rceil$.

Formally,

Algorithm 1 PTAS for Covering Integer Program

```
ẑ ← ∞
k ← min(n, ⌈m(1 + ϵ)/ϵ⌉)
for S ⊆ N with |S| ≤ k do
  if ∑_{j∈N\S} a_ij + ∑_{j∈T(S)} a_ij ≥ 1 for all i ∈ [m] then
    obtain a basic optimal solution x^B(S) to LP(S)
    x^I_j(S) ← ⌈x^B_j(S)⌉ for j ∈ N
    z^I(S) ← ∑_{j∈N} c_j x^I_j
    if ẑ > z^I(S) then
      ẑ ← z^I(S), x̂ ← x^I(S)
    end if
  end if
end for
output ẑ as the value and x̂ as the certificate.
```

Theorem 4.2. *The above algorithm is a $1+\epsilon$ approximation for CIP.*

Proof. Let x^* be an optimal solution to (4.1) with the value z^*. Let $S^* = \{j | x_j = 0\}$. If $|S^*| \leq k$ then $\hat{z} \leq z^*$ which by optimality of z^* implies $\hat{z} = z^*$.

Otherwise, let $S^* = \{i_1, i_2, \ldots, i_r\}$ such that $c_1 \leq c_2 \leq \ldots c_r$. Let $S_k^* = \{i_1, \ldots, i_k\}$ and $\sigma = \sum_{j \in S_k} c_j$. Observe that if $j \in N \setminus (S_k^* \cup T(S_k^*))$ then $c_j \leq \sigma/k$.

Now,

$$\begin{aligned} z^* &\geq \sum_{j=1}^{n} x_j^B(S_k^*) && \text{[since } z^* \text{ is also a solution of IP}(S_k^*)] \\ &\geq \sum_{j=1}^{n} \lceil x_j^B(S_k^*) \rceil - \delta \end{aligned}$$

where,

$$\delta = \sum_{j \in D} c_j \quad \text{with} \quad D = \{ j \in N | 0 < x_j^B(S_k^*) < 1 \}.$$

$|D| \leq m$, since $j \in D$ implies x_j is a basic variable in $x^B(S)$. Also, as $D \cap (S_k^* \cup T(S_k^*)) = \emptyset$, so $j \in D$ implies $c_j \leq \sigma/k$. Therefore, $\delta \leq m\sigma/k$ and

$$z^* \geq \hat{z} - m\sigma/k \geq \hat{z} - m\hat{z}/k.$$

□

Notice we need to solve $\mathcal{O}(n^k)$ LPs in the above algorithm. Since each LP can be solved in polynomial time w.r.t. n the above algorithm qualifies as a PTAS but not an EPTAS.

4.4 Geometric Min-Knapsack

In *the 2-dimensional geometric min-knapsack*, 2DGMK, given a set of items I with each item i having a height $h_i \in (0, 1]$, width $w_i \in (0, 1]$, and cost $c_i > 0$, the aim is to obtain subset $I' \subseteq I$ and map each item in $i \in I'$ to a rectangular region $R_i = [l_i, l_i + w_i] \times [b_i, b_i + h_i]$ (or $R_i = [l_i, l_i + h_i] \times [b_i, b_i + w_i]$ if rotations are allowed) such that $\sum_{i \in I'} c_i$ is minimized while satisfying the constraint

$$\bigcup_{i \in I'} R_i \supseteq [0, 1]^2 \tag{4.4}$$

We get the *cardinality 2DGMK* if $c_i = 1$ for all items. If subset I' along with the mapping R_i satisfy (4.4) then I' along with R_i is called a *covering*. In this section we show a reduction from the partition problem to feasibility of 2DGMK without rotation.

Given an instance $I = \{a_1, \ldots, a_n\}$ of partition construct an instance $J = \left\{ \left(\frac{1}{2}, \frac{2a_i}{\sum_{j \in [n]} a_j} \right) | i \in [n] \right\}$.

Observation 4.1. *The total area of rectangles in J is 1.*

Lemma 4.5. *If there is a solution for I then there is a mapping R_i such that $\bigcup_{i \in J} R_i = [0, 1]^2$.*

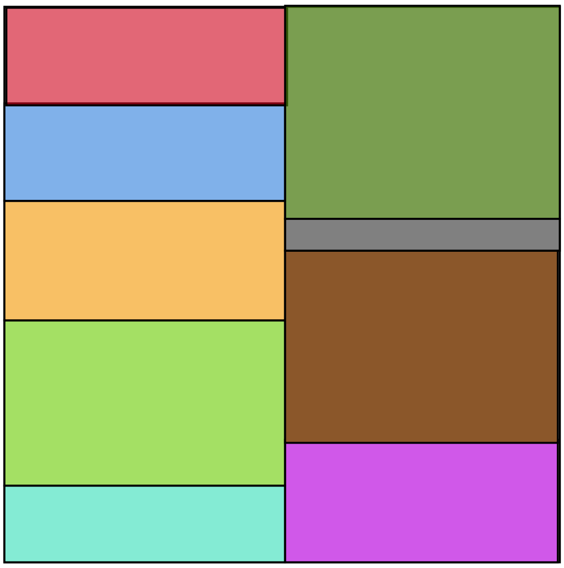


Figure 4.1: 2-dimensional geometric min-knapsack feasibility solves partition

Proof. Suppose there is a partition of I into two sets $I_1 = \{a_{i_1}, \ldots, a_{i_m}\}, I_2 = \{a_{j_1}, \ldots, a_{j_{n-m}}\}$ such that

$$\sum_{k\in[m]} a_{i_k} = \sum_{k\in[n-m]} a_{j_k} = \frac{1}{2}\sum_{i\in[n]} a_i.$$

Let

$$R_{i_k} = [0, 1/2] \times \left[\sum_{k'\in[k-1]} a_{i_{k'}}, \sum_{k'\in[k]} a_{i_{k'}}\right]$$

for $k \in [m]$ and

$$R_{j_k} = [1/2, 1] \times \left[\sum_{k'\in[k-1]} a_{j_{k'}}, \sum_{k'\in[k]} a_{j_{k'}}\right]$$

for $k \in [n - m]$.

Clearly, we have $\bigcup_{k\in[m]} R_{i_k} = [0, 1/2] \times [0, 1]$ and $\bigcup_{k\in[n-m]} R_{j_k} = [1/2, 1] \times [0, 1]$. Hence, $\bigcup_{i\in[n]} R_i = [0, 1]^2$. □

Now we show if there is covering then the rectangles form two columns of width $\frac{1}{2}$.

Lemma 4.6. *If J has a covering consisting of items J' along with the mapping R_i then for any $i \in [n]$ either $R_i = [0, 1/2] \times [a, b]$ or $R_i = [1/2, 1] \times [a, b]$.*

Proof. Suppose J has a covering consisting of items J' along with the mapping R_i. By Observation 4.1 $J = J'$, $\bigcup_{i\in j} r_i \setminus [0, 1]^2$ has 0 area, and $R_i \cap R_j$ has area 0 for any $i, j \in J$. Fix an item

$i_0 \in J$ and let $R_i = [a, a + 1/2] \times [c, d]$. $a > 1/2$ is not possible as that would mean $R_{i_0} \setminus [0, 1]^2$ has area $(d-c)(a-0.5)$ which is non-zero which means $\bigcup_{i\in J} r_i \setminus [0, 1]^2$ has a non-zero area. Say $a \in (0, 1/2)$ then since every item has length $1/2$ any item j covering the region $[0, a] \times [c, d]$ must either have an overlap of non-zero area with i_0 or $R_j \setminus [0, 1]^2$ has non-zero area. □

Finally, we show how to obtain a partition from a covering.

Lemma 4.7. *If there is covering for J then there is a partition for I.*

Proof. Suppose J has a covering consisting of items J' along with the mapping R_i. Again by Observation 4.1 $J = J'$, $\bigcup_{i\in J} r_i \setminus [0, 1]^2$ has 0 area, and $R_i \cap R_j$ has area 0. By Lemma 4.6 we can partition J into $l = \{i|\exists a, b, R_i = [0, 1/2] \times [a, b]\}$ and $r = \{i|\exists a, b, R_i = [1/2, 1] \times [a, b]\}$. Area under items in l must be at least $1/2$ as for any item i, $R_i \cap [0, 1/2] \times [0, 1]$ has 0 area. Similarly area under items in r must be at least $1/2$. But, total area of J is 1. Therefore, $\text{area}(l) = \text{area}(r) = 1/2$, which implies $\sum_{i\in l} a_i = \sum_{i\in r} a_i = \frac{1}{2} \sum_{i\in[n]} a_i$. □

The following theorem follows from Lemma 4.5 and Lemma 4.7.

Theorem 4.3. *Feasibility of 2-dimensional geometric min-knapsack is NP-hard.*

This implies that there does not even exist a polynomial-time approximation algorithm for 2-dimensional geometric min-knapsack.

Chapter 5

Conclusions

In the original proof of the non-existence of an APTAS for vector bin packing, it was claimed that the reduction used was an approximation preserving reduction, and hence, the APX-hardness of the vector bin packing problem was a simple corollary. This is no longer the case now as we could only show that our reduction is a gap preserving reduction, and hence, it is not known if 2-dimensional vector bin packing is indeed APX-hard. More importantly, there is still a considerable gap between the best-known approximation ratio for $d = 2$ case (1.406) and our lower bound (1.00167). We also make similar observation in case of the vector bin covering problem, i.e., we have not showed that it is APX-hard, and the gap between the best-known algorithm for the two-dimensional case (2) and our lower bound (1.001). As Sandeep observed, his reduction (in [41]) does not give the exact lower bounds on the approximation ratio. Instead, it shows that the approximation ratio can not be $o(\ln d)$. Hence, the problem of finding the optimal lower bound on the approximation ratio in the exact sense is still open for vector bin packing with fixed dimension. Also, note that in case of vector bin covering there is a gap of factor $c \log \log d$ between the lower bound and upper bound on the approximation ratio. We observe that Lemma 3.1 implies that the intermediate reduction to 4-Partition is still approximation preserving, and hence, 4-Partition is indeed APX-hard. This also leads us to believe that it will be insightful to look at the problem of improving the bound for 4-Partition (currently the same as MAX-3-DM) with the ultimate aim of improving the lower bound for 2-dimensional vector bin packing and 2-dimensional vector bin covering. For geometric bin packing we do not know any hardness result which grows with d but the best algorithm has an approximation ratio which is exponential in d. As we noted in the Section 1.2, the literature on geometric bin covering is extremely sparse and no algorithms are known for the same.

Our results on covering integer problem are optimal as we show a PTAS along with the non-existence of EPTAS. Since, we were able to show that feasibility of geometric min-knapsack without rotation is NP-hard, therefore only hope of solving the problem is to either use resource augmentation or look at a more restricted version.

Bibliography

[1] Noga Alon, Yossi Azar, János Csirik, Leah Epstein, Sergey V. Sevastianov, Arjen P. A. Vestjens, and Gerhard J. Woeginger. On-line and off-line approximation algorithms for vector covering problems. *Algorithmica*, 21(1):104–118, 1998. doi: 10.1007/PL00009203. URL `https://doi.org/10.1007/PL00009203`. 5

[2] Susan F. Assmann, David S. Johnson, Daniel J. Kleitman, and Joseph Y.-T. Leung. On a dual version of the one-dimensional bin packing problem. *J. Algorithms*, 5(4): 502–525, 1984. doi: 10.1016/0196-6774(84)90004-X. URL `https://doi.org/10.1016/0196-6774(84)90004-X`. 5

[3] Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 13–25. SIAM, 2014. doi: 10.1137/1.9781611973402.2. URL `https://doi.org/10.1137/1.9781611973402.2`. 5

[4] Nikhil Bansal, José R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Math. Oper. Res.*, 31(1):31–49, 2006. doi: 10.1287/moor.1050.0168. URL `https://doi.org/10.1287/moor.1050.0168`. 4, 12, 13, 15, 16

[5] Nikhil Bansal, Alberto Caprara, and Maxim Sviridenko. A new approximation method for set covering problems, with applications to multidimensional bin packing. *SIAM J. Comput.*, 39(4):1256–1278, 2009. doi: 10.1137/080736831. URL `https://doi.org/10.1137/080736831`. 4

[6] Nikhil Bansal, Marek Eliáš, and Arindam Khan. Improved approximation for vector bin packing. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January*

10-12, 2016, pages 1561–1579. SIAM, 2016. doi: 10.1137/1.9781611974331.ch106. URL https://doi.org/10.1137/1.9781611974331.ch106. 3, 4

[7] Piotr Berman and Marek Karpinski. Improved approximation lower bounds on small occurrence optimization. *Electron. Colloquium Comput. Complex.*, 10(008), 2003. URL http://eccc.hpi-web.de/eccc-reports/2003/TR03-008/index.html. 13

[8] Alberto Caprara. Packing 2-dimensional bins in harmony. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 490–499. IEEE Computer Society, 2002. doi: 10.1109/SFCS.2002.1181973. URL https://doi.org/10.1109/SFCS.2002.1181973. 4

[9] Alberto Caprara, Andrea Lodi, and Michele Monaci. An approximation scheme for the two-stage, two-dimensional bin packing problem. In William J. Cook and Andreas S. Schulz, editors, *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, May 27-29, 2002, Proceedings*, volume 2337 of *Lecture Notes in Computer Science*, pages 315–328. Springer, 2002. doi: 10.1007/3-540-47867-1_23. URL https://doi.org/10.1007/3-540-47867-1_23. 4

[10] Chandra Chekuri and Sanjeev Khanna. On multidimensional packing problems. *SIAM J. Comput.*, 33(4):837–851, 2004. doi: 10.1137/S0097539799356265. URL https://doi.org/10.1137/S0097539799356265. 4

[11] Miroslav Chlebík and Janka Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006. doi: 10.1016/j.tcs.2005.11.029. URL https://doi.org/10.1016/j.tcs.2005.11.029. 13

[12] Miroslav Chlebík and Janka Chlebíková. Hardness of approximation for orthogonal rectangle packing and covering problems. *J. Discrete Algorithms*, 7(3):291–305, 2009. doi: 10.1016/j.jda.2009.02.002. URL https://doi.org/10.1016/j.jda.2009.02.002. 4, 6, 7, 12, 13, 20

[13] Janka Chlebíková and Miroslav Chlebík. Hardness of asymptotic approximation for orthogonal rectangle packing and covering problems. *Electron. Colloquium Comput. Complex.*, (019), 2006. URL http://eccc.hpi-web.de/eccc-reports/2006/TR06-019/index.html. 12, 20

[14] Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Comput. Sci. Rev.*,

24:63–79, 2017. doi: 10.1016/j.cosrev.2016.12.001. URL https://doi.org/10.1016/j.cosrev.2016.12.001. 7

[15] Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 262–273. IEEE Computer Society, 1997. doi: 10.1109/CCC.1997.612321. URL https://doi.org/10.1109/CCC.1997.612321. 11

[16] Pierluigi Crescenzi and Alessandro Panconesi. Completeness in approximation classes. *Inf. Comput.*, 93(2):241–262, 1991. doi: 10.1016/0890-5401(91)90025-W. URL https://doi.org/10.1016/0890-5401(91)90025-W. 10

[17] János Csirik, J. B. G. Frenk, Martine Labbé, and Shuzhong Zhang. Two simple algorithms for bin covering. *Acta Cybern.*, 14(1):13–25, 1999. URL https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3507. 5

[18] János Csirik, David S. Johnson, and Claire Kenyon. Better approximation algorithms for bin covering. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pages 557–566. ACM/SIAM, 2001. URL http://dl.acm.org/citation.cfm?id=365411.365533. 5

[19] Wenceslas Fernandez de la Vega and George S. Lueker. Bin packing can be solved within $1+\epsilon$ in linear time. *Comb.*, 1(4):349–355, 1981. doi: 10.1007/BF02579456. URL https://doi.org/10.1007/BF02579456. 3

[20] Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995. doi: 10.1016/0304-3975(94)00097-3. URL https://doi.org/10.1016/0304-3975(94)00097-3. 25

[21] Alan M Frieze and MRB Clarke. Approximation algorithms for the m-dimensional 0-1 knapsack problem: worst-case and probabilistic analyses. *European Journal of Operational Research*, 15(1):100–109, 1984. 6, 24, 26

[22] Waldo Gálvez, Fabrizio Grandoni, Sandy Heydrich, Salvatore Ingala, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via l-packings. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 260–271. IEEE Computer Society, 2017. doi: 10.1109/FOCS.2017.32. URL https://doi.org/10.1109/FOCS.2017.32. 7

[23] M. R. Garey, Ronald L. Graham, and Jeffrey D. Ullman. Worst-case analysis of memory allocation algorithms. In Patrick C. Fischer, H. Paul Zeiger, Jeffrey D. Ullman, and Arnold L. Rosenberg, editors, *Proceedings of the 4th Annual ACM Symposium on Theory of Computing, May 1-3, 1972, Denver, Colorado, USA*, pages 143–150. ACM, 1972. doi: 10.1145/800152.804907. URL https://doi.org/10.1145/800152.804907. 2

[24] Fabrizio Grandoni, Stefan Kratsch, and Andreas Wiese. Parameterized approximation schemes for independent set of rectangles and geometric knapsack. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 53:1–53:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/LIPIcs.ESA.2019.53. URL https://doi.org/10.4230/LIPIcs.ESA.2019.53. 7

[25] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi: 10.1145/502090.502098. URL https://doi.org/10.1145/502090.502098. 9

[26] Rebecca Hoberg and Thomas Rothvoss. A logarithmic additive integrality gap for bin packing. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2616–2625. SIAM, 2017. doi: 10.1137/1.9781611974782.172. URL https://doi.org/10.1137/1.9781611974782.172. 3

[27] Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975. doi: 10.1145/321906.321909. URL https://doi.org/10.1145/321906.321909. 2, 6

[28] Klaus Jansen and Lars Prädel. New approximability results for two-dimensional bin packing. *Algorithmica*, 74(1):208–269, 2016. doi: 10.1007/s00453-014-9943-z. URL https://doi.org/10.1007/s00453-014-9943-z. 5

[29] Klaus Jansen and Roberto Solis-Oba. An asymptotic fully polynomial time approximation scheme for bin covering. *Theor. Comput. Sci.*, 306(1-3):543–551, 2003. doi: 10.1016/S0304-3975(03)00363-3. URL https://doi.org/10.1016/S0304-3975(03)00363-3. 5

[30] Klaus Jansen and Guochuan Zhang. Maximizing the number of packed rectangles. In Torben Hagerup and Jyrki Katajainen, editors, *Algorithm Theory - SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings*, volume 3111 of *Lecture Notes in Computer Science*, pages 362–371. Springer, 2004. doi:

10.1007/978-3-540-27810-8_31. URL https://doi.org/10.1007/978-3-540-27810-8_31. 7

[31] Klaus Jansen and Guochuan Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007. doi: 10.1007/s00453-006-0194-5. URL https://doi.org/10.1007/s00453-006-0194-5. 7

[32] Viggo Kann. Maximum bounded 3-dimensional matching is MAX snp-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991. doi: 10.1016/0020-0190(91)90246-E. URL https://doi.org/10.1016/0020-0190(91)90246-E. 13

[33] Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 312–320. IEEE Computer Society, 1982. doi: 10.1109/SFCS.1982.61. URL https://doi.org/10.1109/SFCS.1982.61. 3

[34] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9. 1

[35] Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. *Inf. Process. Lett.*, 110(16):707–710, 2010. doi: 10.1016/j.ipl.2010.05.031. URL https://doi.org/10.1016/j.ipl.2010.05.031. 6, 25

[36] Ariel Kulik, Hadas Shachnai, Oded Shmueli, and Robert Sayegh. Approximation schemes for deal splitting and covering integer programs with multiplicity constraints. *Theor. Comput. Sci.*, 412(52):7087–7098, 2011. doi: 10.1016/j.tcs.2011.09.018. URL https://doi.org/10.1016/j.tcs.2011.09.018. 24

[37] Eugene L. Lawler. Fast approximation algorithms for knapsack problems. *Math. Oper. Res.*, 4(4):339–356, 1979. doi: 10.1287/moor.4.4.339. URL https://doi.org/10.1287/moor.4.4.339. 2, 6

[38] Joseph Y.-T. Leung, Tommy W. Tam, C. S. Wong, Gilbert H. Young, and Francis Y. L. Chin. Packing squares into a square. *J. Parallel Distributed Comput.*, 10(3):271–275, 1990.

doi: 10.1016/0743-7315(90)90019-L. URL https://doi.org/10.1016/0743-7315(90)90019-L. 7

[39] Erez Petrank. The hardness of approximation: Gap location. *Comput. Complex.*, 4:133–157, 1994. doi: 10.1007/BF01202286. URL https://doi.org/10.1007/BF01202286. 13

[40] Thomas Rothvoß. Approximating bin packing within $O(\log \mathrm{OPT} \cdot \log\log \mathrm{OPT})$ bins. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 20–29. IEEE Computer Society, 2013. doi: 10.1109/FOCS.2013.11. URL https://doi.org/10.1109/FOCS.2013.11. 3

[41] Sai Sandeep. Almost optimal inapproximability of multidimensional packing problems. *Electron. Colloquium Comput. Complex.*, 28:7, 2021. URL https://eccc.weizmann.ac.il/report/2021/007. 4, 6, 12, 31

[42] Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001. ISBN 978-3-540-65367-7. URL http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-65367-7. 9

[43] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. ISBN 978-0-521-19527-0. URL http://www.cambridge.org/de/knowledge/isbn/item5759340/?site_locale=de_DE. 9

[44] Gerhard J. Woeginger. There is no asymptotic PTAS for two-dimensional vector packing. *Inf. Process. Lett.*, 64(6):293–297, 1997. doi: 10.1016/S0020-0190(97)00179-8. URL https://doi.org/10.1016/S0020-0190(97)00179-8. ii, 12, 13, 14, 15, 19, 20