Computer Science Review 24 (2017) 63–79

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Survey

Approximation and online algorithms for multidimensional bin packing: A survey[☆]

Henrik I. Christensen[a], Arindam Khan[b,*,1], Sebastian Pokutta[c], Prasad Tetali[c]

[a] University of California, San Diego, USA

[b] Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), Scuola universitaria professionale della Svizzera italiana (SUPSI), Università della Svizzera italiana (USI), Switzerland

[c] Georgia Institute of Technology, Atlanta, USA

ARTICLE INFO

Article history:
Received 13 August 2016
Received in revised form
23 November 2016
Accepted 20 December 2016
Available online 16 January 2017

Keywords:
Approximation algorithms
Online algorithms
Multidimensional packing and covering
Bin packing
Multidimensional scheduling
Geometric packing

ABSTRACT

The bin packing problem is a well-studied problem in combinatorial optimization. In the classical bin packing problem, we are given a list of real numbers in (0, 1] and the goal is to place them in a minimum number of *bins* so that no bin holds numbers summing to more than 1. The problem is extremely important in practice and finds numerous applications in scheduling, routing and resource allocation problems. Theoretically the problem has rich connections with discrepancy theory, iterative methods, entropy rounding and has led to the development of several algorithmic techniques. In this survey we consider approximation and online algorithms for several classical generalizations of bin packing problem such as geometric bin packing, vector bin packing and various other related problems. There is also a vast literature on mathematical models and exact algorithms for bin packing. However, this survey does not address such exact algorithms.

In two-dimensional GEOMETRIC BIN PACKING, we are given a collection of rectangular items to be packed into a minimum number of unit size square bins. This variant has a lot of applications in cutting stock, vehicle loading, pallet packing, memory allocation and several other logistics and robotics related problems. In d-dimensional VECTOR BIN PACKING, each item is a d-dimensional vector that needs to be packed into unit vector bins. This problem is of great significance in resource constrained scheduling and in recent virtual machine placement in cloud computing. We also consider several other generalizations of bin packing such as geometric knapsack, strip packing and other related problems such as vector scheduling, vector covering etc. We survey algorithms for these problems in offline and online setting, and also mention results for several important special cases. We briefly mention related techniques used in the design and analysis of these algorithms. In the end we conclude with a list of open problems.

© 2016 Elsevier Inc. All rights reserved.

Contents

1. Introduction 64
1.1. Organization of the Survey 65
2. Preliminaries 65
2.1. Approximation algorithms and inapproximability 65
2.2. One dimensional bin packing 66
2.2.1. Offline 1-D bin packing 66

☆ This research was supported by NSF EAGER award grants CCF-1415496 and CCF-1415498.

* Corresponding author.
E-mail addresses: hichristensen@ucsd.edu (H.I. Christensen), arindam.khan@supsi.ch (A. Khan), sebastian.pokutta@isye.gatech.edu (S. Pokutta), tetali@math.gatech.edu (P. Tetali).

1 Supported by SNF Grant 200021_159697/1. A part of this work was done when the author was at Georgia Institute of Technology.

http://dx.doi.org/10.1016/j.cosrev.2016.12.001
1574-0137/© 2016 Elsevier Inc. All rights reserved.

2.2.2. Online 1-D bin packing 67
2.3. Multidimensional bin packing 67
2.3.1. Geometric packing 67
2.3.2. Vector packing 67
2.3.3. Relation between the problems 68
2.4. Techniques 68
2.4.1. NextFitDecreasingHeight (NFDH) 68
2.4.2. Steinberg's Packing 68
2.4.3. Configuration LP 69
2.4.4. Algorithms based on rounding items to constant number of types 69
2.4.5. Round and Approx (R&A) Framework 70
3. Geometric bin packing 70
3.1. Geometric bin packing 70
3.2. Online packing 71
3.3. Square packing 71
3.4. Resource augmentation 71
3.5. Strip packing 71
3.6. Shelf and guillotine packing 72
3.7. Geometric knapsack 73
3.8. Other related problems 73
4. Vector bin packing 73
4.1. Offline vector packing 74
4.2. Online vector packing 74
4.3. Vector knapsack 75
4.4. Vector scheduling 75
4.5. Vector bin covering 75
5. Open problems 75
References 77

1. Introduction

The bin packing problem has been the corner stone of approximation algorithms and has been extensively studied starting from the early seventies. In the classical BIN PACKING problem, we are given a list $I = \{i_1, i_2, \ldots, i_n\}$ of real numbers in the range $(0, 1]$, the goal is to place them in a minimum number of *bins* so that no bin holds numbers summing to more than 1.

Bin packing is a special case of the one-dimensional CUTTING STOCK problem [1], loading problem [2] and several scheduling related problems [3]. One of the first nontrivial results on bin packing was due to Ullman in 1971 [4]. He studied the problem from the standpoint of memory allocation problems such as table formatting, prepaging and file allocation, and noticed that finding a general placement algorithm for attaining the minimum number of bins appears to be impractical. He then provided asymptotic analysis of two heuristics: FirstFit (FF) and BestFit (BF). Soon in 1972, Garey, Graham and Ullman [5], provided more detailed analysis of four heuristics: FirstFit, BestFit, FirstFitDecreasing (FFD) and BestFitDecreasing (BFD). In his thesis, Johnson [6] studied several other algorithms and analysis techniques for bin packing problems. Then Johnson, Demers, Ullman, Garey and Graham [7] published definitive analysis of the worst case guarantees of several bin packing approximation algorithms. The bin packing problem is well-known to be NP-hard [8] and the seminal work of Johnson et al. initiated an extremely rich research area in approximation algorithms [9]. In fact the term *approximation algorithm* was coined by David S. Johnson [10] in an influential and prescient paper in 1973 where he studied algorithms for bin packing and other packing and covering related optimization problems. The introductory chapter of *Handbook of Approximation Algorithms and Metaheuristics* [11] mentions: "*Research on bin packing problem and its variants has attracted very talented investigators who have generated more than 650 papers, most of which deal with approximations*". For online algorithms, bin packing (and related load balancing problem) is one of the key problems. In the book *Online Computation and Competitive Analysis* [12], bin packing has been used as the first introductory example to explain online algorithms.

Bin packing is also extremely useful in practice and has a lot of applications in various fields. Skiena [13] has presented *market research* for the field of combinatorial optimization and algorithms, attempting to determine which algorithmic problems are most in demand for applications, by studying WWW traffic. Both bin packing and related knapsack problem were among top five most popular NP-hard problems. The implementations of algorithms for bin packing and knapsack problems were the most needed among all NP-hard problems, even more than problems such as set-cover, traveling salesman and graph-coloring.

Garey and Johnson [14], followed by Coffman et al. [15], gave comprehensive surveys on bin packing algorithms. Coffman and Lueker also covered probabilistic analyses of packing algorithms in detail [16]. Galambos and Woeginger [17] gave an overview restricted mainly to online variants of bin packing problems. There had been many surveys on bin packing problems thereafter [11,18,19]. The most recent, extensive coverage on 1-D bin packing was given by Coffman et al. [20].

In this survey, we primarily focus on packing in higher dimensions due to its prominence in many real world applications. We primarily consider two generalizations of bin packing: GEOMETRIC BIN PACKING and VECTOR BIN PACKING.

In two-dimensional (2-D) GEOMETRIC BIN PACKING (GBP), we are given a collection of rectangular items to be packed into a minimum number of unit-size square bins. This variant and other higher dimensional GBP variants have vast applications in cutting stock, vehicle loading, pallet packing, memory allocation and several other logistics and robotics related problems [1,21]. In two dimensions, packing objects into containers has many important applications, e.g., in the context of cutting out a given set of patterns from a given large piece of material minimizing waste, typically in sheet metal processing and apparel fabrication. In three dimensions, these problems are frequently encountered in minimizing storage space or container space for transportation. In this survey we consider the widely studied *orthogonal packing* case, where the items must be placed in the bin such that their sides are parallel to the sides of the bin. In any feasible solution, items are not allowed to overlap. Here two variants are usually studied,

(i) where the items cannot be rotated (packing by *translations*), and (ii) they can be rotated by 90 degrees (packing by *restricted rigid* motions). These variants are also recurrent in practice, e.g., in apparel production usually there are patterns of weaving or texture on the material so that the position where a piece should be cut cannot be rotated arbitrarily.

In *d*-dimensional VECTOR BIN PACKING (VBP), each item is a *d*-dimensional vector that needs to be packed into unit vector bins. The problem is of great significance in resource constrained scheduling and appeared also recently in virtual machine placement in cloud computing [22]. For example, consider each job (item) has multiple resource requirements (dimensions) such as CPU, memory, I/O, disk, network etc. and each server (bin) has a bounded amount of these resources. The goal to assign all jobs to minimum number of servers, without violating the resource constraints, translates to the vector packing problem. Even in two dimensions, vector packing has many novel applications in layout design, logistics, loading and scheduling problems [23,24].

These generalizations have been well studied since the 1970s. Baker, Coffman, and Rivest first considered orthogonal packings in two dimensions [25]. At the same time Coffman et al. [26] gave performance bounds for level-oriented two-dimensional packing algorithms such as NEXTFITDECREASINGHEIGHT (NFDH) and FIRSTFITDECREASINGHEIGHT (FFDH). Lodi, Martello and Monaci first gave a survey on two-dimensional packing problems [27]. Epstein and van Stee gave a survey in [11] on multi-dimensional bin packing. There has been consistent progress in the area since then. We will provide a detailed survey of related works in the later corresponding sections.

1.1. Organization of the Survey

In Section 2, we introduce related definitions and notation. In Section 3, we discuss results related to geometric bin packing. Thereafter in Section 4, we discuss results related to vector packing. Finally, in Section 5 we conclude with a list of open problems.

2. Preliminaries

In this section we introduce relevant notation and definitions required to define, analyze and classify bin packing related problems. Some more additional definitions will be introduced later on as required.

2.1. Approximation algorithms and inapproximability

Approximation Algorithm is an attempt to systematically measure, analyze, compare and improve the performance of heuristics for intractable problems. It gives theoretical insight on how to find fast solutions for practical problems, provides mathematical rigor to study and analyze heuristics, and also gives a metric for the difficulty of different discrete optimization problems. Let Π be a *minimization* problem, $\mathcal{I}$ be the set of input instances for Π and $\mathcal{A}$ be an algorithm for Π. Let $\mathcal{A}(I)$ be the objective function value of the solution returned by algorithm $\mathcal{A}$ on instance I and $\mathrm{Opt}(I)$ be the objective function value of the corresponding optimal solution.

Definition 2.1 (*Approximation Ratio*). $\mathcal{A}$ is a ρ-(multiplicative)-approximation algorithm if $\mathcal{A}(I) \le \rho \cdot \mathrm{Opt}(I)$ holds for every input instance $I \in \mathcal{I}$.

The infimum over the set of all values ρ such that $\mathcal{A}$ is ρ-approximation is called to be the approximation ratio $\rho_{\mathcal{A}}$ of the algorithm.

In this survey we consider problems for which $\mathrm{Opt}(I) > 0$ where alternatively we can also define:

$$\rho_{\mathcal{A}} = \sup_{I \in \mathcal{I}} \left\{ \frac{\mathcal{A}(I)}{\mathrm{Opt}(I)} \right\}.$$

Analogously an algorithm $\mathcal{A}$ for a *maximization* problem Π' is called a ρ-approximation algorithm if $\mathcal{A}(I) \ge \frac{1}{\rho} \cdot \mathrm{Opt}(I)$ holds for every instance I of Π'. This asymmetry ensures that $\rho \ge 1$ for all approximation algorithms.

In some cases, quality of the heuristic is measured in terms of additive approximation. In other words, an algorithm $\mathcal{A}$ for a minimization problem Π is called a σ-additive approximation algorithm if $\mathcal{A}(I) \le \mathrm{Opt}(I) + \sigma$ holds for every instance I of Π. Additive approximation algorithms are relatively rare. Karmarkar–Karp's algorithm [28] for one-dimensional bin packing is one such example.

For detailed introduction to approximation algorithms, we refer the readers to the following books on approximation algorithms [29,30,11,9].

Definition 2.2 (*Polynomial Time Approximation Scheme* (PTAS)). A problem is said to admit a *polynomial time approximation scheme* (PTAS) if for every constant $\varepsilon > 0$, there is a *poly*(n)-time algorithm with approximation ratio $(1+\varepsilon)$ where n is the size of the input. Here running time can be as bad as $O(n^{f(1/\varepsilon)})$ for any function f that depends only on ε.

If the running time of PTAS is $O(f(1/\varepsilon) \cdot n^c)$ for some function f and a constant c that is independent of ε, we call it to be an *efficient polynomial time approximation scheme* (EPTAS).

On the other hand, if the running time of PTAS is polynomial in both n and $1/\varepsilon$, it is said to be a *fully polynomial time approximation scheme* (FPTAS).

Assuming P $\neq$ NP, a PTAS is the best result we can obtain for a strongly NP-hard problem. Already in the 1-D case, a simple reduction from the PARTITION problem shows that it is NP-hard to determine whether a set of items can be packed into two bins or not, implying that no approximation better than 3/2 is possible. However, this does not rule out the possibility of an Opt + 1 guarantee where Opt is the number of bins required in the optimal packing. Hence it is insightful to consider the *asymptotic approximation ratio*.

Definition 2.3 (*Asymptotic Approximation Ratio* (AAR)). The *asymptotic approximation ratio* $\rho^{\infty}_{\mathcal{A}}$ of an algorithm $\mathcal{A}$ is:

$$\lim_{n\to\infty} \sup \rho^n_{\mathcal{A}}, \quad \text{where } \rho^n_{\mathcal{A}} = \sup_{I \in \mathcal{I}} \left\{ \frac{\mathcal{A}(I)}{\mathrm{Opt}(I)} \middle| \mathrm{Opt}(I) = n \right\}.$$

In this context the approximation ratio defined as in Definition 2.1, is also called to be the (*absolute*) *approximation ratio*.

Definition 2.4 (*Asymptotic PTAS* (APTAS)). A problem is said to admit an *asymptotic polynomial time approximation scheme* (APTAS) if for every $\varepsilon > 0$, there is a poly-time algorithm with asymptotic approximation ratio of $(1+\varepsilon)$.

If the running time of APTAS is polynomial in both n and $1/\varepsilon$, it is said to be an *asymptotic fully polynomial time approximation scheme* (AFPTAS).

Note that NP optimization problems whose decision versions are all polynomial time reducible to each other (due to NP-completeness), can behave very differently in their approximability. For example classical bin packing problem admits an APTAS, whereas no polynomial factor approximation is known for the traveling salesman problem. This anomaly is due to the fact that reductions between NP-complete problems preserve polynomial time computability, but not the quality of the approximate solution.

PTAS is the class of problems that admit polynomial time approximation scheme. On the other hand, APX is the class of problems that have a constant-factor approximation. A problem Π_1 is said to be APX-hard if there is a PTAS-reduction from every problem $\Pi_2 \in$ APX to Π_1, and to be APX-complete if Π_1 is APX-hard and also in APX. In practice, reducing one problem to another to demonstrate APX-completeness is often done using other reduction schemes, such as *L*-reductions and *E*-reductions, which imply PTAS reductions. We refer the readers to Chapter 15 of [11] for more details on these complexity classes and related notion of reductions. Note that PTAS $\subseteq$ APX. In fact the containment is strict unless P = NP.

Theorem 2.5 ([31]). *If a problem* $\mathscr{F}$ *is* APX*-hard then it does not admit a* PTAS *unless* P = NP.

In fact there is no quasi-polynomial time approximation scheme (QPTAS) for any APX-hard problem unless NP $\subseteq$ QP.

Online algorithms: Optimization problems where the input is received in an online manner (i.e., the input does not arrive as a single batch but as a sequence of input portions) and the output must be produced online (i.e., the system must react in response to each incoming portion taking into account that future input is not known at any point in time) are called online problems. Bin packing is also one of the key problems in online algorithms. Let us define the notion of a competitive ratio which will be useful when we discuss some related results in online algorithms in later sections.

Definition 2.6 (*Competitive Ratio*). An online algorithm $\mathcal{A}$ for a *minimization* problem Π is called *c-competitive* if there exists a constant δ such that for all finite input sequences I, $\mathcal{A}(I) \le c \cdot \mathrm{Opt}(I) + \delta$. If the additive constant $\delta \le 0$, we say $\mathcal{A}$ to be *strictly c-competitive*.

The infimum over the set of all values c such that $\mathcal{A}$ is c-competitive is called the *competitive ratio* of $\mathcal{A}$.

We will sometimes call the above defined competitive ratio to be *asymptotic competitive ratio* and strictly competitive ratio to be *absolute competitive ratio*. In general, there are no requirements or assumptions concerning the computational efficiency of an online algorithm. However, in practice, we usually seek polynomial time online algorithms. We refer the readers to [32,12] for more details on online algorithms.

There are few others metrics to measure the quality of a packing, such as *random-order ratio* [33], *accommodation function* [34], *relative worst-order ratio* [35], *differential approximation measure* [36] etc.

2.2. One dimensional bin packing

Before going to multidimensional bin packing, we give a brief description of the results in 1-D bin packing. Here we focus primarily on very recent results. For a detailed survey and earlier results we refer the interested reader to [20].

2.2.1. Offline 1-D bin packing

The earliest algorithms for one dimensional (1-D) bin packing were simple greedy algorithms such as FirstFit (FF), NextFit (NF), FirstFitDecreasing (FFD), NextFitDecreasing (NFD) etc. In their celebrated work, de la Vega and Lueker [37] gave the first APTAS by introducing linear grouping that reduces the number of different item types. Algorithms based on other item grouping or rounding based techniques have been used in many related problems. The result was substantially improved by Karmarkar and Karp [28] who gave a guarantee of $\mathrm{Opt} + O(\log^2 \mathrm{Opt})$ by providing an iterative rounding for a linear programming formulation. It was then improved by Rothvoß [38] to $\mathrm{Opt} + O(\log \mathrm{Opt} \cdot \log\log \mathrm{Opt})$ using ideas from discrepancy theory. Very recently, Hoberg and Rothvoß [39] achieved an approximation with $\mathrm{Opt} + O(\log \mathrm{Opt})$ bins using discrepancy method coupled with a novel 2-stage packing approach. On the other hand, the possibility of an algorithm with an Opt + 1 guarantee is still open. This is one of the top ten open problems in the field of approximation algorithms mentioned in [30].

Table 1 summarizes different algorithms and their performance guarantees. Here $T_\infty \approx 1.69103$ is the well-known *harmonic* constant that appears ubiquitously in the context of bin packing. It is defined as follows: $T_\infty = \sum_{i=1}^{\infty} \frac{1}{t_i - 1}$ where $t_1 = 2, t_{i+1} = t_i(t_i - 1) + 1$ for $i \ge 1$.

The *Gilmore–Gomory LP relaxation* [1] is used in [37,28,38] to obtain better approximation. This LP is of the following form:

$$\min\left\{\mathbf{1}^T x | Ax = \mathbf{1}, x \ge 0\right\}. \quad (1)$$

Here A is the pattern matrix that consists of all column vectors $\{p \in \mathbb{N}^n | p^T s \le \mathbf{1}\}$ where $s := (s_1, s_2, \ldots, s_n)$ is the size vector for the items. Each such column p is called a pattern and corresponds to a feasible multiset of items that can be assigned to a single bin. Now if we only consider patterns $p \in \{0, 1\}^n$, LP (1) can be interpreted as an LP relaxation of a set cover problem, in which a set I of items has to be covered by *configurations* from the collection $\mathcal{C} \subseteq 2^I$, where each configuration $C \in \mathcal{C}$ corresponds to a set of items that can be packed into a bin:

$$\min\left\{\sum_{C \in \mathcal{C}} x_C : \sum_{C \ni i} x_C \ge 1 \ (i \in I), x_C \in \{0, 1\} \ (C \in \mathcal{C})\right\}. \quad (2)$$

This configuration LP is also used in other algorithms for multidimensional bin packing and we will continue the discussion of configuration LPs in later sections.

Let Opt and Opt_f be the value of the optimal integer solution and fractional solution for LP (1) respectively. Although LP (1) has an exponential number of variables, one can compute a basic solution x with $\mathbf{1}^T x \le \mathrm{Opt}_f + \delta$ in time polynomial in n and $1/\delta$ using the Grötschel–Lovász–Schrijver variant of the Ellipsoid method [44] or the Plotkin–Shmoys–Tardos framework [45,46]. In fact the analysis of [39], only shows an upper bound of $O(\log \mathrm{Opt})$ on the additive integrality gap of LP (1). It has been conjectured in [47] that the LP has the *Modified Integer Roundup Property*, i.e., $\mathrm{Opt} \le \lceil \mathrm{Opt}_f \rceil + 1$. The conjecture has been proved true for the case when the instance contains at most 7 different item sizes [48]. Recently, Eisenbrand et al. [49] found a connection between coloring permutations and bin packing, that shows that *Beck's Three Permutation Conjecture* (any three permutations can be bi-colored with O(1) discrepancy) would imply a $O(1)$ integrality gap for instances with all items sizes bigger than 1/4. However, Newman, Neiman, and Nikolov [50] found a counterexample to Beck's conjecture. Using these insights Eisenbrand et al. [49] showed that a broad class of algorithms cannot give an $o(\log n)$ gap. Rothvoß [51] further explored the connection to discrepancy theory and gave a rounding using Beck's entropy method achieving an $O(\log^2 \mathrm{Opt})$ gap alternatively. The later improvement to $O(\log \mathrm{Opt})$ in [38,39] arose from the constructive partial coloring lemma [52] and gluing techniques. Recently Goemans and Rothvoß [53] also have shown polynomiality for bin packing when there are $O(1)$ number of item types. Very recently, Jansen and Klein [54] showed that bin packing is in FPT, parameterized by the number of vertices of an underlying integer knapsack polytope.

Bin packing problem is also well-studied when the number of bins is some fixed constant k. If the sizes of the items

Table 1
Approximation algorithms for one dimensional bin packing.

Algorithm	Performance guarantee	Techniques
NextFit [7]	$2 \cdot \text{Opt}$	Greedy, online,
NextFitDecreasing [7]	$T_\infty \cdot \text{Opt} + O(1)$ [40]	Presorting
FirstFit [7]	$\lfloor 1.7\text{Opt} \rfloor$ [41]	Greedy, online
FirstFitDecreasing [7]	$\frac{11}{9}\text{Opt} + \frac{6}{9}$ [42]	Presorting
BestFit [7]	$\lfloor 1.7\text{Opt} \rfloor$ [43]	Greedy, online
de la Vega and Lueker [37]	$(1+\varepsilon)\text{Opt} + O(\frac{1}{\varepsilon^2})$	Linear grouping
Karp and Karmarkar [28]	$\text{Opt} + O(\log^2 \text{Opt})$	Iterative rounding
Rothvoß [38]	$\text{Opt} + O(\log \text{Opt} \cdot \log\log \text{Opt})$	Discrepancy methods
Hoberg and Rothvoß [39]	$\text{Opt} + O(\log \text{Opt})$	Discrepancy methods

are polynomially bounded integers, then the problem can be solved *exactly* using dynamic programming in $n^{O(k)}$ time, for an input of length n. Along with APTAS for bin packing, this implies a pseudo-polynomial time approximation scheme for bin packing, significantly better than 3/2, the hardness of (absolute) approximation for the problem. However, Jansen et al. [55] showed *unary bin packing* (where item sizes are given in unary encoding) is W[1]-hard and thus the running time for fixed number of bins k, cannot be improved to $f(k) \cdot n^{O(1)}$ for any function of k, under standard complexity assumptions.

2.2.2. Online 1-D bin packing

An online bin packing algorithm uses *k-bounded space* if, for each item, the choice of where to pack it, is restricted to a set of at most k open bins. Lee and Lee [56] gave a $O(1)$*-bounded space* algorithm Harmonic that achieve asymptotic competitive ratio of $T_\infty \approx 1.69$. They also established tightness. For unbounded space, there had been a series of improvements for the asymptotic competitive ratio: RefinedHarmonic (competitive ratio: $\frac{373}{228} < 1.63597$) [56], ModifiedHarmonic (competitive ratio <1.61562) and ModifiedHarmonic2 (competitive ratio <1.61217) [57], Harmonic++ (competitive ratio <1.58889) [58]. Ramanan et al. [57] showed that harmonic-type algorithms cannot achieve better than 1.58333 asymptotic competitive ratio. Very recently, this lower bound was beaten by Heydrich and van Stee [59] who presented an online algorithm SonOfHarmonic with asymptotic competitive ratio of at most 1.5815 using a new type of *interval classification*. They also gave a lower bound of 1.5766 for any interval classification algorithm. In general the best known lower bound for asymptotic competitive ratio is $248/161 \approx 1.54037$ [60], improved from previous best 1.54014 [61]. Very recently, Balogh et al. [62] presented an online bin packing algorithm with an *absolute* competitive ratio of 5/3 which is optimal. We refer the readers to [63] for more variants (such as dynamic bin packing, bin packing with rejections, bin packing with cardinality constraints, resource augmentation etc.) and techniques related to online bin packing.

Online bin packing has also been studied in probabilistic settings. Shor [64] gave tight-bounds for average-case online bin packing. Other related algorithms for online stochastic bin packing are *Sum of Squares* algorithm by Csirik et al. [65] and primal–dual based algorithms in [66].

2.3. Multidimensional bin packing

We will now discuss preliminaries related to multidimensional bin packing. We will consider the offline setting, where all items are known a priori. We also briefly survey results in the online setting, when the items appear one at a time and we need to decide packing an item without knowing the future items.

2.3.1. Geometric packing

Definition 2.7 (*Two-Dimensional Geometric Bin Packing (2-D GBP)*)**.** In two-dimensional geometric bin packing (2-D GBP), we are given a collection of n rectangular items $I := \{r_1, r_2, \ldots, r_n\}$ where each rectangle r_k is specified by its width and height (w_k, h_k) such that w_k, h_k are rational numbers in $(0, 1]$. The goal is to pack all rectangles into a minimum number of unit square bins.

We consider the widely studied *orthogonal packing* case, where the items must be placed in the bin such that their sides are parallel to the sides of the bin. In any feasible solution, items are not allowed to overlap. Here two variants are usually studied, (i) where the items cannot be rotated, and (ii) they can be rotated by 90°.

We will also mention some results related to strip packing and geometric knapsack problems, two other geometric generalizations of bin packing, in Section 3.

Definition 2.8 (*Strip Packing (2-D SP)*)**.** In two-dimensional strip packing (2-D SP), we are given a strip of unit width and infinite height, and a collection of n rectangular items $I := \{r_1, r_2, \ldots, r_n\}$ where each rectangle r_k is specified by its width and height (w_k, h_k) such that w_k, h_k are rational numbers in $(0, 1]$. The goal is to pack all rectangles into the strip minimizing the height.

Strip packing also has a lot of applications in industrial engineering and computer science. Recently, there have been a lot of applications of strip packing in electricity allocation and peak demand reductions in smart grids [67–69].

Definition 2.9 (*Geometric Knapsack (2-D GK)*)**.** In two-dimensional geometric knapsack (2-D GK), we are given a unit square bin and a collection of two dimensional rectangles $I := \{r_1, r_2, \ldots, r_n\}$ where each rectangle r_k is specified by its width and height (w_k, h_k) and profit p_k such that w_k, h_k, p_k are rational numbers in $(0, 1]$. The goal is to find the maximum profit subset that can be feasibly packed into the bin.

Multidimensional variants of above three geometric problems are defined analogously using d-dimensional rectangular parallelepipeds (also known as d-orthotope, the generalization of rectangles in higher dimensions) and d-dimensional cuboids (also known as d-cube, the generalization of squares in higher dimensions). We will discuss 3-dimensional variants in more detail in Section 3.

2.3.2. Vector packing

Now we define vector bin packing, the nongeometric generalization of bin packing.

Definition 2.10 (*Vector Bin Packing (d-D VBP)*)**.** In d-dimensional vector packing (d-D VBP), we are given a set of n rational vectors $I := \{v_1, v_2, \ldots, v_n\}$ from $[0, 1]^d$. The goal is to partition them into sets (bins) $B_1, B_2, \ldots, B_m$ such that $\|\sigma_{B_j}\|_\infty \le 1$ for $1 \le j \le m$ where $\sigma_{B_j} = \sum_{v_i \in B_j} v_i$ is the sum of vectors in B_j, and we want to minimize m, the number of bins.

In other words, the goal is to pack all the vectors into minimum number of bins so that for every bin the sum of packed vectors in the bin should not exceed the vector of the bin in each dimension.

We now define related vector knapsack, vector scheduling and vector bin covering problems.

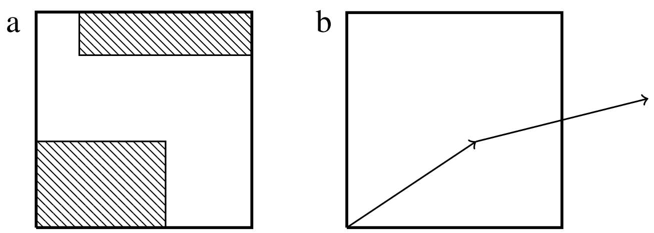


Fig. 1. Two rectangles of size $\frac{3}{5} \times \frac{2}{5}$ and $\frac{4}{5} \times \frac{1}{5}$ can be packed into one unit square bin. However, vectors $(\frac{3}{5}, \frac{2}{5})$ and $(\frac{4}{5}, \frac{1}{5})$ cannot be packed into $(1, 1)$ vector bin.

Definition 2.11 (*Vector Knapsack (d-D VK)*)**.** In d-dimensional VECTOR KNAPSACK (d-D VK), we are given a vector bin with capacity $(b_1, b_2, \ldots, b_d)$ and a collection of n rational vectors $I := \{v_1, v_2, \ldots, v_n\}$ from $[0, 1]^d$ and profit $p_i > 0$ for each vector v_i. The goal is to find a maximum profit subset S that can be feasibly packed into the bin, i.e., total size of items in S is bounded by b_k for each dimension $k \in [d]$.

For $d = 1$, this is the classical knapsack problem.

Definition 2.12 (*Vector Scheduling (d-D VS)*)**.** In d-dimensional VECTOR SCHEDULING (d-D VS), we are given a set of n rational vectors $I := \{v_1, v_2, \ldots, v_n\}$ from $[0, 1]^d$ and an integer m. The goal is to partition I into m sets $B_1, B_2, \ldots, B_m$ such that $\max_{1 \le i \le m} \|\sigma_{B_i}\|_\infty$ is minimized, where $\sigma_{B_i} = \sum_{v_i \in B_i} v_i$ is the sum of vectors in B_i.

For $d = 1$, this just reduces to the classical multiprocessor scheduling.

Definition 2.13 (*Vector Bin Covering (d-D VBC)*)**.** In d-dimensional VECTOR BIN COVERING (d-D VBC), we are given a set of n rational vectors $I := \{v_1, v_2, \ldots, v_n\}$ from $[0, 1]^d$. The goal is to partition them into sets (bins) $B_1, B_2, \ldots, B_m$ such that $\sigma_{B_j} \ge 1$ in all dimensions for all $j \in [m]$, where $\sigma_{B_j} = \sum_{v_i \in B_j} v_i$ is the sum of vectors in B_j, and we want to maximize m, the number of bins.

For $d = 1$, bin covering problem admits APTAS [70]. Vector bin covering is sometimes also called *dual bin packing* or *dual vector packing* in the literature.

2.3.3. *Relation between the problems*

Fig. 1(a) and (b) show the difference between geometric packing and vector packing. Given a set of vectors, one can easily determine whether they can be packed into one unit bin by just checking whether the sum along each coordinate is at most one or not. However for geometric bin packing, it is NP-hard to determine whether a set of rectangles can be packed into one unit square bin or not, implying that no (absolute) approximation better than 2 is possible even for 2-D GBP.

Note that both geometric knapsack and strip packing are closely related to geometric bin packing. Results and techniques related to strip packing and knapsack have played a major role in improving the approximation for geometric bin packing. If all items have same height then d-dimensional strip packing reduces to $(d-1)$-dimensional geometric bin packing. On the other hand to decide whether a set of rectangles (w_i, h_i) for $i \in [n]$ can be packed into m bins, one can define a 3-D geometric knapsack instance with n items $(w_i, h_i, 1/m)$ and profit $(w_i \cdot h_i \cdot 1/m)$ and decide if there is a feasible packing with profit $\sum_{i \in n}(w_i \cdot h_i \cdot 1/m)$. Fig. 2 shows the relation between different generalizations of bin packing.

There are few other generalizations of bin packing such as weighted bipartite edge coloring. We do not cover them here and we refer the readers to [71,72].

2.4. *Techniques*

Now we describe some techniques, heavily encountered in multidimensional packing.

2.4.1. NEXTFITDECREASINGHEIGHT *(NFDH)*

Among all the heuristics for geometric packing, NEXTFITDECREASINGHEIGHT (NFDH) is one of the most used heuristics in the literature. We will discuss several key properties of NFDH that makes it widely useful. This procedure was introduced by Coffman et al. [26] for 2-D geometric packing, though Meir and Moser already used a similar variant for rectangle packing in 1968. NFDH considers items in a non-increasing order of height and greedily assigns items in this order into *shelves*, where a *shelf* is a row of items having their bases on a line that is either the base of the bin or the line drawn at the top of the highest item packed in the shelf below. More specifically, items are packed left-justified starting from bottom-left corner of the bin, until the next item cannot be included. Then the shelf is closed and the next item is used to define a new shelf whose base touches the tallest (left most) item of the previous shelf. If the shelf does not fit into the bin, the bin is closed and a new bin is opened. The procedure continues until all the items are packed. This simple heuristic works quite well for small items. Some key properties of NFDH are following:

Lemma 2.14 ([73])**.** *Let B be a rectangular region with width w and height h. We can pack small rectangles (with both width and height less than ε) with total area A using NFDH into B if $w \ge \varepsilon$ and $w \cdot h \ge 2A + w^2/8$.*

Lemma 2.15 ([74])**.** *Given a set of items of total area of V and each having height at most one, they can be packed in at most $4V + 3$ unit bins by NFDH.*

Lemma 2.16 ([26])**.** *Let B be a rectangular region with width w and height h. If we pack small rectangles (with both width and height less than ε) using NFDH into B, total $w \cdot h - (w + h) \cdot \varepsilon$ area can be packed, i.e., the total wasted volume in B is at most $(w + h) \cdot \varepsilon$.*

In fact it can be generalized to d-dimensions.

Lemma 2.17 ([75])**.** *Let C be a set of d-dimensional cubes (where $d \ge 2$) of sides smaller than ε. Consider NFDH heuristic applied to C. If NFDH cannot place any other cube in a rectangle R of size $r_1 \times r_2 \times \cdots r_d$ (with $r_i \le 1$), the total wasted (unfilled) volume in that bin is at most: $\varepsilon \sum_{i=1}^{d} r_i$.*

2.4.2. *Steinberg's Packing*

Steinberg [76] provided a 2-approximation for strip packing by the following result:

Theorem 2.18 (*A. Steinberg [76]*)**.** *We are given a set of items I' and a bin Q of size $w \times h$. Let $w_{\max} \le w$ and $h_{\max} \le h$ be the maximum width and maximum height among the items in I' respectively and $a(I')$ be the total area of the rectangles in I'. Also we denote $x_+ := \max(x, 0)$. If*

$$2a(I') \le wh - (2w_{\max} - w)_+(2h_{\max} - h)_+$$

then I' can be packed into Q in polynomial time.

This algorithm has wide applicability in many other geometric packing problems as it gives a packing of any set of rectangles into a bin if their total area satisfies the above constraint.

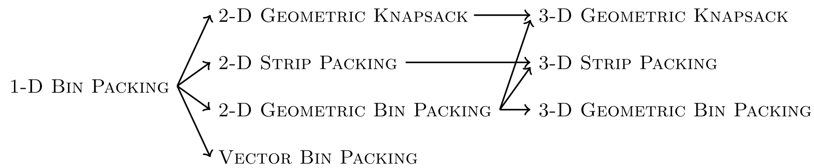


Fig. 2. Generalizations of bin packing.

2.4.3. Configuration LP

The best known approximations for most bin packing type problems are based on strong LP formulations called configuration LPs. Here, there is a variable for each possible way of feasibly packing a bin (called a *configuration*). This allows the packing problem to be cast as a set covering problem, where each item in the instance I must be covered by some configuration. Let $\mathcal{C}$ denote the set of all valid configurations for the instance I. The configuration LP is defined as:

$$\min\left\{\sum_{C\in\mathcal{C}} x_C : \sum_{C\ni i} x_C \geq 1\ \forall i\in I,\ x_C \geq 0\ \forall C\in\mathcal{C}\right\}. \quad (3)$$

As the size of $\mathcal{C}$ can possibly be exponential in the size of I, one typically considers the dual of the LP given by:

$$\max\left\{\sum_{i\in I} v_i : \sum_{i\in C} v_i \leq 1\ \forall C\in\mathcal{C},\ v_i \geq 0\ \forall i\in I\right\}. \quad (4)$$

The separation problem for the dual is the following knapsack problem. Given set of weights v_i, is there a feasible configuration with total weight of items more than 1. From the well-known connection between separation and optimization [77,45,44], solving the dual separation problem to within a $(1+\varepsilon)$ accuracy suffices to solve the configuration LP within $(1+\varepsilon)$ accuracy. We refer the readers to [51] for an explicit proof that, for any set family $\mathcal{C} \subseteq 2^{[n]}$, if the dual separation problem can be approximated to $(1+\varepsilon)$-factor in time $T(n,\varepsilon)$ then the corresponding column-based LP can be solved within an arbitrarily small additive error δ in time $poly(n, \frac{1}{\delta}) \cdot T(n, \Omega(\frac{\delta}{n}))$. This error term cannot be avoided as otherwise we can decide the PARTITION problem in polynomial time. For 1-D BP, the dual separation problem admits an FPTAS, i.e., it can be solved in time $T(n,\varepsilon) = poly(n, \frac{1}{\varepsilon})$. Thus the configuration LP can be solved within arbitrarily small additive constant error δ in time $poly(n, \frac{1}{\delta}) \cdot poly(n, O(\frac{n}{\delta}))$. For multidimensional vector bin packing, the dual separation problem admits a PTAS, i.e., can be solved in time $T(n,\varepsilon) = O(n^{f(\frac{1}{\varepsilon})})$. Thus the configuration LP can be solved within $(1+\delta)$ multiplicative factor in time $poly(n, \frac{1}{\delta}) \cdot T(n, \Omega(\frac{\delta n}{n}))$, i.e., in time $O(n^{O(f(\frac{1}{\delta}))})$.

Note that the configurations in (3) are defined based on the original item sizes (without any rounding). However, for more complex problems (say 3-D GBP) one cannot hope to solve such an LP to within $(1+\varepsilon)$ (multiplicative) accuracy, as the dual separation problem becomes at least as hard as 2-D GBP. In general, given a problem instance I, one can define a configuration LP in multiple ways (say where the configurations are based on rounded sizes of items in I, which might be necessary if the LP with original sizes is intractable). For the special case of 2-D GBP, the separation problem for the dual (4) is the 2-D geometric knapsack problem for which the best known result is only a 2-approximation. However, Bansal et al. [78] showed that the configuration LP (3) with original sizes can still be solved to within $1+\varepsilon$ accuracy (this is a non-trivial result and requires various ideas).

Similarly for the case of vector bin packing, the separation problem for the dual (4) is the vector knapsack problem which can be solved to within $1+\varepsilon$ accuracy [79]. However, there is no EPTAS even for 2-D vector knapsack [80].

The fact that solving the configuration LP does not incur any loss for 2-D GBP or VBP plays a key role in the present best approximation algorithms.

2.4.4. Algorithms based on rounding items to constant number of types

Rounding of items to $O(1)$ types has been used either implicitly [81] or explicitly [37,28,82–84], in almost all bin packing algorithms to reduce the problem complexity and make it tractable. Let I be a given set of items and s_x be the size of item $x \in I$. Define a partial order on bin packing instances as follows: $I \leq J$ if there exists a bijective function $f : I \to J$ such that $s_x \leq s_{f(x)}$ for each item $s \in I$. J is then called a rounded up instance of I. One of the key properties of rounding items is as follows:

Lemma 2.19 (*[28]*). *$I \leq J$ implies* $\mathrm{Opt}(I) \leq \mathrm{Opt}(J)$.

There are typically two types of rounding: either the size of an item in some coordinate (such as width or height) is rounded in an instance-oblivious way (e.g., in harmonic rounding [56,82], or rounding sizes to geometric powers [28]), or it is rounded in an input sensitive way (e.g., in linear grouping [37]).

Linear grouping: Linear grouping was introduced by Fernandez de la Vega and Lueker [37] in the first approximation scheme for 1-D bin packing problem. It is a technique to reduce the number of distinct item sizes. The scheme works as follows, and is based on a parameter k, to be fixed later. Sort the n items by nonincreasing size and partition them into $\lceil 1/k \rceil$ groups such that the first group consists of the largest $\lceil nk \rceil$ pieces, next group consists of the next $\lceil nk \rceil$ largest items and so on, until all items have been placed in a group. Apart from the last group all other groups contain $\lceil nk \rceil$ items and the last group contains $\leq nk$ items. The rounded instance $\tilde{I}$ is created by discarding the first group, and for each other group, the size of an item is rounded up to the size of the largest item in that group. The following lemma shows that the optimal packing of these rounded items is very close to the optimal packing of the original items.

Lemma 2.20 (*[37]*). *Let $\tilde{I}$ be the set of items obtained from an input I by applying linear grouping with group size $\lceil nk \rceil$, then*

$$\mathrm{Opt}(\tilde{I}) \leq \mathrm{Opt}(I) \leq \mathrm{Opt}(\tilde{I}) + \lceil nk \rceil$$

and furthermore, any packing of $\tilde{I}$ can be used to generate a packing of I with at most $\lceil nk \rceil$ additional bins.

If all items are $>\varepsilon$, then $n\varepsilon < \mathrm{Opt}$. So, for $k = \varepsilon^2$ we get that any packing of $\tilde{I}$ can be used to generate a packing of I with at most $\lceil n\varepsilon^2 \rceil \leq n\varepsilon^2 + 1 < \varepsilon \cdot \mathrm{Opt} + 1$ additional bins. See [71] for a slightly modified version of linear grouping that does not lose the additive constant of 1.

Geometric grouping: Karmarkar and Karp [28] introduced a refinement of linear grouping called geometric grouping with parameter k. Let $\alpha(I)$ be the smallest item size of an instance I. For $r = 0, 1, \ldots, \lfloor \log_2 \frac{1}{\alpha(I)} \rfloor$, let I_r be the instance consisting of items $i \in I$ such that $s_i \in (2^{-(r+1)}, 2^{-r}]$. Let J_r be the instances obtained by applying linear grouping with parameter $k \cdot 2^r$ to I_r. If $J = \cup_r J_r$ then:

Lemma 2.21 (*[28]*). $\mathrm{Opt}(J) \le \mathrm{Opt}(I) \le \mathrm{Opt}(J) + k\lceil \log_2 \frac{1}{\alpha(I)} \rceil$.

Harmonic rounding: Lee and Lee [56] introduced a Harmonic algorithm (*harmonic*$_k$) for online 1-D bin packing, where each item j with $s_j \in (\frac{1}{q+1}, \frac{1}{q}]$ for $q \in \{1, 2, \dots, (k-1)\}$, is rounded to $1/q$. Then q items of type $1/q$ can be packed together in a bin. So for each type q, we open one bin B_q and only items of type q are packed into that bin and a new bin for type q items is opened when B_q is full. Let $t_1 = 1$, $t_{q+1} := t_q(t_q + 1)$ for $q \ge 1$. Let $m(k)$ be the integer with $t_{m(k)} < k < t_{m(k)+1}$. It is shown in [56] that the asymptotic approximation ratio of *harmonic*$_k$ is $T_k = \sum_{q=1}^{m(k)} \frac{1}{t_q} + \frac{1}{t_{m(k)+1}} \cdot \frac{k}{k-1}$. When $k \to \infty$, $T_\infty = 1.691\dots$, this is the harmonic constant, ubiquitous in bin packing. Caprara [82] introduced the HarmonicDecreasingHeight algorithm for 2-D GBP with asymptotic approximation ratio of T_∞, where widths are rounded according to harmonic rounding and then same width items are packed using NFDH. We refer the readers to [85] for more related applications of the Harmonic algorithm in online and bounded space algorithms.

2.4.5. Round and Approx (R&A) Framework

Now we describe the R&A Framework as described in [86]. It is the key framework used to obtain present best approximation algorithms for 2-D geometric bin packing and vector bin packing. It uses a ρ-approximation algorithm that rounds the items to $O(1)$ types, as a subroutine to obtain $(1 + \ln \rho)$ approximation. The framework is described as follows:

1. Solve the LP relaxation of (3) using the APTAS ([78] for 2-D GBP, [79] for VBP). Let x^* be the (near)-optimal solution of the LP relaxation and let $z^* = \sum_{C \in \mathcal{C}} x^*_C$. Let r the number of configurations in the support of x^*.
2. Initialize a $|\mathcal{C}|$-dimensional binary vector x^r to be an all-0 vector. For $\lceil (\ln \rho) z^* \rceil$ iterations repeat the following: select a configuration $C' \in \mathcal{C}$ at random with probability $x^*_{C'}/z^*$ and let $x^r_{C'} := 1$.
3. Let S be the remaining set of items not covered by x^r, i.e., $i \in S$ if and only if $\sum_{C \ni i} x^r_C = 0$. On set S, apply the ρ-approximation algorithm $\mathcal{A}$ that rounds the items to $O(1)$ types and then pack. Let x^a be the solution returned by $\mathcal{A}$ for the residual instance S.
4. Return $x = x^r + x^a$.

Let $\mathrm{Opt}(S)$ and $\mathcal{A}(S)$ denote the value of the optimal solution and the approximation algorithm used to solve the residual instance, respectively. Since the algorithm uses randomized rounding in step 2, the residual instance S is not known in advance. However, the algorithm should perform "well" independent of S. For this purpose, Bansal, Caprara and Sviridenko [86] defined the notion of *subset-obliviousness* where the quality of the approximation algorithm to solve the residual instance is expressed using a small collection of vectors in $\mathbb{R}^{|I|}$.

Definition 2.22. An asymptotic ρ-approximation for the set covering problem defined in (1), is called subset-oblivious if, for any fixed $\varepsilon > 0$, there exist constants k, Λ, β (possibly dependent on ε), such that for every instance I of (1), there exist vectors $v^1, v^2, \dots, v^k \in \mathbb{R}^{|I|}$ that satisfy the following properties:

1. $\sum_{i \in C} v^j_i \le \Lambda$, for each configuration $C \in \mathcal{C}$ and $j = 1, 2, \dots, k$;
2. $\mathrm{Opt}(I) \ge \sum_{i \in I} v^j_i$ for $j = 1, 2, \dots, k$;
3. $\mathcal{A}(S) \le \rho \sum_{i \in S} v^j_i + \varepsilon \mathrm{Opt}(I) + \beta$, for each $S \subseteq I$ and $j = 1, 2, \dots, k$.

Roughly speaking, the vectors are analogues to the sizes of items and are introduced to use the properties of the dual of (1). Property 1 says that the vectors divided by constant Λ must be feasible for (2). Property 2 provides lower bound for $OPT(I)$ and Property 3 guarantees that the $\mathcal{A}(S)$ is not significantly larger than ρ times the lower bound in Property 2 associated with S.

The main result about the R&A is the following.

Theorem 2.23 (*Simplified*). *If a problem has a asymptotic ρ-approximation algorithm that is subset oblivious, and the configuration LP with original item sizes can be solved to within $(1+\varepsilon)$ accuracy in polynomial time for any $\varepsilon > 0$, then the R&A framework gives a $(1 + \ln \rho)$-asymptotic approximation.*

Very recently, Bansal and Khan [87] extended the R&A framework to any constant rounding based algorithms for 2-D GBP. Then Bansal, Elias and Khan [88] further showed that any constant rounding based algorithm for VBP is also subset-oblivious.

One can derandomize the above procedure and get a deterministic version of R&A method in which Step 2 is replaced by a greedy procedure that defines x^r guided by a suitable potential function. See [86] for the details of derandomization.

3. Geometric bin packing

In this section we give an extensive survey of the literature related to geometric packing and other related problems.

3.1. Geometric bin packing

Two-dimensional geometric bin packing (GBP) is substantially different from the 1-D case. Bansal et al. [75] showed that 2-D bin packing in general does not admit an APTAS unless P = NP.

On the positive side, there has also been a long sequence of works giving improved approximation algorithms. We refer readers to [27] for a review of several greedy heuristics such as NextFitDecreasing, FirstFitDecreasing, BestFitDecreasing, FiniteBestStrip, FloorCeiling, FiniteFirstFit, KnapsackPacking, FiniteBottomLeft, AlternateDirections etc. For the case when we do not allow rotation, until the mid 90's the best known bound was a 2.125 approximation [89], which was improved by Kenyon and Rémila [90] to a $2 + \varepsilon$ approximation (this follows as a corollary of their APTAS for 2-D strip packing) for any $\varepsilon > 0$. Caprara in his break-through paper [82] gave an algorithm for 2-D bin packing attaining an asymptotic approximation of $T_\infty (\approx 1.69103)$. For the case when rotations are allowed, Miyazawa and Wakabayashi [91] gave an algorithm with an asymptotic performance guarantee of 2.64. Epstein and Stee [92] improved it to 2.25 by giving a packing where, in almost all bins, an area of 4/9 is occupied. Finally an asymptotic approximation guarantee arbitrarily close to 2 followed from the result of [93]. This was later improved by Bansal et al. [86] to $(\ln(T_\infty) + 1) \approx 1.52$, for both the cases with and without rotation, introducing a novel randomized rounding based framework: *Round and Approx (R & A)* framework. Jansen and Prädel [83] improved this guarantee further to give a 1.5-approximation algorithm. Their algorithm is based on exploiting several non-trivial structural properties of how items can be packed in a bin.

Very recently, Bansal and Khan [87] gave a polynomial time algorithm with an asymptotic approximation guarantee of $\ln(1.5) + 1 \approx 1.405$ for 2-D GBP. This is the best algorithm known so far, and holds both for the case with and without rotations. The main idea behind this result is to show that the *Round and Approx* (R&A) framework introduced by Bansal, Caprara and Sviridenko [86] (See Section 2.4.5) can be applied to the $(1.5 + \varepsilon)$-approximation result of Jansen and Prädel [83]. They give a more general argument to apply the R&A framework directly to a wide class of algorithms. In particular, they show that *any* algorithm based on rounding the (large) items into $O(1)$ types, is subset-oblivious. The main observation is that any ρ-approximation based on rounding the item sizes, can be related to another configuration LP (on rounded item sizes) whose solution is no worse than ρ times the optimum solution. They also give some results to show the limitations of rounding based algorithms in obtaining better approximation ratios. There are typically two types of rounding: either the size of an item in some coordinate (such as width or height) is rounded up in an instance-oblivious way (e.g., harmonic rounding [56,82], or geometric rounding [28]), or it is rounded up in an input sensitive way (e.g. linear grouping [37]). They show that any rounding based algorithm that rounds at least one side of each large item to some number in a constant-size collection values chosen independent of problem instance (let us call such rounding *input-agnostic*), cannot have an approximation ratio better than 3/2. For arbitrary constant rounding based algorithms they show a hardness of 4/3.

We remark that there is still a huge gap between these upper bounds and known lower bounds. In particular, the best known explicit lower bound on the asymptotic approximation for 2-D BP is currently $1 + 1/3792$ and $1 + 1/2196$, for the versions with and without rotations, respectively [94]. The best asymptotic worst-case ratio that is achievable in polynomial time for d-dimensional GBP for $d > 2$ is T_∞^{d-1} [82], and in fact it can be achieved by an online algorithm using bounded space. There are no known explicit better lower bounds for higher dimensions.

In *non-asymptotic setting*, without rotations there are 3-approximation algorithms by Zhang [95] and also by Harren and van Stee [96]. Harren and van Stee [97] gave a non-asymptotic 2-approximation with rotations. Independently this approximation guarantee is also achieved for the version without rotations by Harren and van Stee [98] and Jansen et al. [99]. These 2-approximation results match the non-asymptotic lower bound for this problem, unless P = NP.

3.2. Online packing

Coppersmith and Raghavan [100] first studied online multidimensional GBP and gave algorithms with asymptotic competitive ratio of 3.25 and 6.35 for dimension $d = 2$ and 3 respectively. Csirik and van Vliet [101] gave an algorithm with competitive ratio T_∞^d (This gives 2.859 for 2-D) for arbitrary dimension d. Epstein and van Stee [92] achieved the same ratio of T_∞^d only using bounded space and showed it to be the optimal among all bounded space algorithms. For $d = 2$, there has been a common approach to use two 1-D online bin packing algorithm A and B to get algorithm $A \otimes B$. Intuitively, algorithm A assigns items to *slices* of height 1 and a determined width depending on the width of the item decided by the algorithm. Algorithm B assigns slices to bins according to its type. This resulting algorithm is called $A \times B$. $A \otimes B$ is a randomized algorithm that uses $A \times B$ and $B \times A$ with equal probability. In 2002, Seiden and van Stee [102] proposed an elegant algorithm called $H \otimes C$, comprised of the Harmonic algorithm H and the ImprovedHarmonic algorithm C, for the 2-D online bin packing problem and proved that the algorithm has an asymptotic competitive ratio of at most 2.66013. Han et al. [103] gave an improved upper bound of 2.5545. Their main idea is to develop new weighting functions for Harmonic++ algorithm and propose new techniques to bound the total weight in a rectangular bin. The best known lower bound is 1.907 by Blitz, van Vliet and Woeginger [104], which is unpublished and now lost (See [63]). The previous best result was 1.856 [105]. We refer the readers to a recent article by van Stee [106] for a discussion on these lower bounding techniques.

Fujita and Hada [107] first considered 2-D online bin packing with rotations and gave two algorithms. Epstein [108] gave an improved bounded space algorithm with competitive ratio 2.54679 and showed a lower bound of 2.53536 for bounded space algorithms. For unbounded space, Epstein [109] gave an algorithm with asymptotic competitive ratio of 2.45. Later Epstein and van Stee [110] gave an algorithm with asymptotic competitive ratio of 2.25.

For 3-D online bin packing, calculations based on present techniques are messy. Han et al. [103] mentions their algorithm gives 4.3198-competitive ratio. Blitz, van Vliet and Woeginger claimed a lower bound of 2.111 [104] in their lost manuscript and the penultimate best was 2.043 by van Vliet [105].

3.3. Square packing

Leung et al. [111] have shown that even the special case of packing squares into square is still NP-hard. Kohayakawa et al. [112] gave a $(2 - (2/3)^d + \varepsilon)$ asymptotic approximation for packing d-dimensional cubes into unit cubes. Later Bansal et al. [75] have given an APTAS for the problem of packing d-dimensional cubes into d-dimensional unit cubes.

For the special case where items are squares, there is also a large number of results for online packing. Coppersmith and Raghavan [100] showed their algorithm gives asymptotic competitive ratio of 2.6875 in this case. They also gave a lower bound of 4/3. Seiden and van Stee [102] gave an algorithm with asymptotic competitive ratio of 2.24437. Epstein and van Stee [113] have shown an upper bound of 2.2697 and a lower bound of 1.6406 for online square packing, and an upper bound of 2.9421 and a lower bound of 1.6680 for online cube packing. The upper bound for squares can be further reduced to 2.24437 using a computer-aided proof. Later Han et al. [114] get an upper bound of 2.1187 for square packing and 2.6161 for cube packing. For bounded space online algorithms, Epstein and van Stee [115] showed lower and upper bounds for optimal online bounded space hypercube packing till dimensions 7. In particular, for 2-D it lies in (2.3634, 2.3692) and for 3-D it lies in (2.956, 3.0672). Very recently, Heydrich et al. [116] have improved the lower bound for online square packing to 1.6707. They also show that the Harmonic-type algorithms cannot break the barrier of 2 for $d = 2$ by giving lower bound of 2.02 for that case. For large dimensions, their lower bound tends to 3.

Table 2 summarizes present best approximation/inapproximability results for geometric bin packing. Here OFF denotes offline, ON denotes online, REC denotes rectangles, CUB denotes cubes, WR denotes with rotation and NR denotes without rotation.

3.4. Resource augmentation

Due to pathological worst-case examples, bin packing has been well-studied under *resource augmentation*, i.e., the side length of the bin is augmented to $(1 + \varepsilon)$ instead of one. This is also known as *bin stretching*. Though 2-D GBP does not admit an APTAS, Bansal et al. [75] gave a polynomial time algorithm to pack rectangles into at most m number of bins of size $(1 + \varepsilon) \times (1 + \varepsilon)$ where m is the optimal number of unit bins needed to pack all items. Later Bansal and Sviridenko [118] showed that this is possible even when we relax the size of the bin in only one dimension.

3.5. Strip packing

Strip packing is a natural generalization of the 1-D bin packing problem (when all the rectangles have the same height) and makespan minimization (when all the rectangles have the same width). This problem is also closely tied with the geometric bin packing problem. As we had stated earlier, the best approximation algorithm for 2-D GBP used to be a factor $(2 + \varepsilon)$ and was a corollary from the APTAS for 2-D strip packing due to Kenyon and Rémila [90]. In the 2-D variant, we are given a strip with width one and unlimited height and the goal is to pack 2-D rectangular

Table 2
Present state of the art for geometric bin packing.

Problem	Dim.	Subcase	Best algorithm	Best lower bound
Geometric bin packing	2	OFF-REC-WR	asymp[a]: 1.405 [87]	1+1/3792 [94]
			abs[b]: 2 [97]	2 (folklore)
		OFF-REC-NR	asymp: 1.405 [87]	1+1/2196 [94]
			abs: 2 [98]	2 (folklore)
	d	OFF-REC-NR	asymp: T_∞^{d-1} for $d > 2$ [82]	1+1/2196 [94]
		OFF-CUB	asymp: PTAS [75]	NP-hard
			abs: 2 [75]	2 [117]
	2	ON-REC-NR	asymp: 2.5545 [103]	1.856 [105]
		ON-REC-WR	asymp: 2.25 [110]	1.6707 [116]
		ON-CUB	asymp: 2.1187[114]	1.6707 [116]
	3	ON-REC-NR	asymp: 4.3198 [103]	2.043 [105]
		ON-CUB	asymp: 2.6161 [114]	1.6707 [116]

[a] Here asymp. means asymptotic approximation guarantee.
[b] Here abs. means absolute approximation guarantee.

items into the strip so as to minimize the height of the packing. In three dimensions, we are given 3-D rectangular items each of whose dimensions is at most one and they need to be packed into a single 3-D box of unit depth, unit width and unlimited height so as to minimize the height of the packing.

First let us discuss offline *absolute approximation* algorithms for 2-D strip packing. Baker et al. [25] introduced the problem in 1980 and gave an algorithm with absolute approximation ratio of 3. Later Coffman et al. [26] introduced two *level-oriented* algorithms: NextFitDecreasingHeight (NFDH), FirstFitDecreasingHeight (FFDH) [26], achieving absolute approximation ratio as 3 and 2.7 respectively. Let $h_{\max}$ be the largest height of any rectangle in the input set. Observe that trivially $OPT \geq h_{\max}$. Sleator [119] gave an algorithm that generates packing of height $2OPT + \frac{h_{\max}}{2}$, hence achieving a 2.5 approximation. Afterwards, Steinberg [76] and Schiermeyer [120] independently improved the approximation ratio to 2. Harren and van Stee [98] first broke the barrier of 2 with their 1.9396 approximation. For the absolute approximation, the present best approximation is due to Harren et al. [121] who have given a $(5/3+\varepsilon)$-approximation whereas the lower bound is 3/2 which follows from one dimensional bin packing.

Recently, there has been progress on *pseudo-polynomial time absolute approximation* of strip packing. Nadiradze et al. [122] overcame the 3/2-inapproximability barrier by presenting a $(1.4+\varepsilon)$-absolute approximation algorithm with pseudo-polynomial running time. Then Gálvez, Grandoni, Ingala and Khan [123] pushed approach of [122] to its limit and gave improved approximation of $\frac{4}{3} + \varepsilon$ in pseudo-polynomial time. They also extended the algorithm to the case with rotations and achieved the same approximation factor by using a dynamic program to pack a *container*-based packing. Jansen et al. [124] also independently achieved the same factor. Adamaszek et al. [125] showed that it is NP-hard to approximate strip packing better than 12/11 even for polynomially bounded data. This also shows strip packing admits no QPTAS, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{poly\log(n)})$.

For *asymptotic approximation*, NFDH and FFDH algorithms by Coffman et al. [26], achieve approximations of 2 and 1.7, respectively. After a sequence of improvements [126,127], the seminal work of Kenyon and Rémila [90] provided an APTAS with an additive term $O\left(\frac{h_{\max}}{\varepsilon^2}\right)$ using a nice interplay of techniques like fractional strip packing, linear grouping and a variant of NFDH. The latter additive term was subsequently improved to $h_{\max}$ by Jansen and Solis-Oba [128].

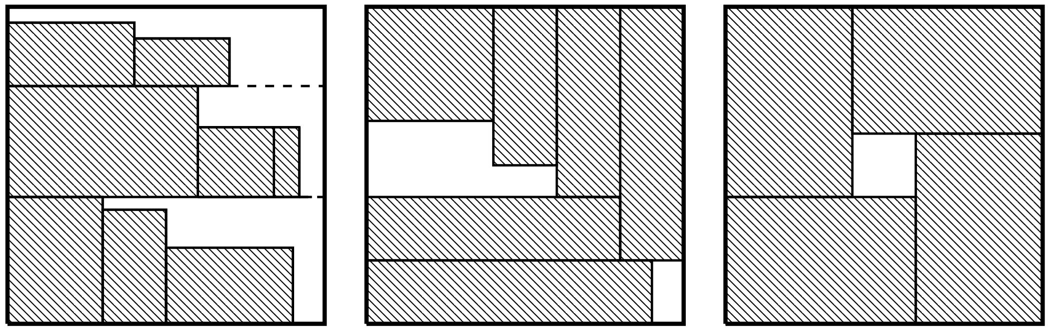
Strip packing *with rotations* is much less studied in the literature. It seems that most techniques that work for the case without rotations can be extended to the case with rotations, however this is not always a trivial task. In particular, it is not hard to achieve a $2 + \varepsilon$ approximation, and the 3/2 hardness of approximation extends to this case as well [128]. In terms of asymptotic approximation, Miyazawa and Wakabayashi [91] gave an algorithm with asymptotic performance ratio of 1.613. Later, Epstein and van Stee [110] gave a $\frac{3}{2}$ asymptotic approximation. Finally, Jansen and van Stee [93] achieved an APTAS for the case with rotations.

Now we discuss *online algorithms* for 2-D strip packing. Baker and Schwartz [129] showed that FirstFit has asymptotic performance ratio 1.7. Csirik and Woeginger [130] improved it to $T_\infty \approx 1.691$ using the Harmonic algorithm as a subroutine. They also mention a lower bound of 1.5401. Recently Han et al. [131] have shown that strip packing achieves the same asymptotic approximation ratio as well as same asymptotic competitive ratio as 1-D BP. For the *absolute competitive ratio*, Brown et al. [132] have given a lower bound of 2. Kern and Paulus [133] gave lower bound of 2.457 with an algorithm with matching competitive ratio for the lower bound input sequence. Later Harren and Kern [134] gave a lower bound of 2.589 and they gave an algorithm with 2.618-competitive ratio for their lower bound example. For the upper bound, Hurink and Paulus [135] and Ye et al. [136] independently achieved the same competitive ratio: $\frac{7}{2} + \sqrt{10} \approx 6.6623$.

3-D strip packing is a common generalization of both the 2-D bin packing problem (when each item has height exactly one) and the 2-D strip packing problem (when each item has width exactly one). Li and Cheng [137] were among the first people who considered the problem. They showed 3-D versions of FFDH and NFDH have unbounded worst-case ratio. They gave a 3.25 approximation algorithm, and later gave an online algorithm with upper bound of $T_\infty^2 \approx 2.89$ [138] using the Harmonic algorithm as a subroutine. Bansal et al. [139] gave a 1.69 asymptotic approximation for the offline case. Recently Jansen and Prädel [140] further improved it to 1.5. Both these two algorithms extend techniques from 2-D bin packing.

3.6. Shelf and guillotine packing

For $d = 2$, many special structures of packings have been considered in the literature, because they are both easy to deal with and important in practical applications. Among these, very famous are the two-stage packing structures, leading to two-dimensional shelf bin packing (2SBP) and two-dimensional shelf strip packing (2SSP). Two-stage packing problems were originally introduced by Gilmore and Gomory [20] and, thinking in terms of cutting instead of packing, requires that each item be obtained from the associated bin by at most two stages of cutting.



(a) Two-stage packing. (b) Guillotine packing. (c) Non-guillotine packing.

Fig. 3. Different types of packing.

In two-stage packing, in the first stage, the bins are horizontally cut into shelves. The second stage produces slices, which contain a single item by cutting the shelves vertically. Finally, an additional stage (called trimming) is allowed in order to separate an item from a waste area. See Fig. 3(a) for an example of two-stage packing. Two-stage packing is equivalent to packing the items into the bins in shelves, where a *shelf* is a row of items having their bases on a line that is either the base of the bin or the line drawn at the top of the highest item packed in the shelf below. Formally, a shelf is a set S of items such that total width $\sum_{j\in S} w_j \le 1$; its height $h(S)$ is given by $\max_{j\in S} h_j$. Many classical heuristics for 2-D strip packing [26,129,130] and 2-D GBP [89], including NFDH and FFDH, construct solutions that are in fact feasible for the two-stage versions. Moreover, Caprara et al. [74] presented an APTAS, for both 2SBP and 2SSP. Given this situation, it is natural to ask for the asymptotic worst-case ratio of general packing versus two-stage packing. Csirik and Woeginger [130] showed ratio of 2SSP versus 2-D strip packing is equal to T_∞. Caprara [82] showed the ratio of 2SBP versus 2-D GBP is also equal to T_∞. Both their algorithms are online and based on HarmonicDecreasingHeight (HDH) heuristic. Now consider the optimal 2SBP solution in which the shelves are horizontal as well as the optimal 2SBP solution in which they are vertical. (Recall that near-optimal 2SBP solutions can be found in polynomial time [74].) There is no evidence that the asymptotic worst-case ratio between the best of these two solutions and the optimal 2-D GBP can be as bad as T_∞, and in fact Caprara conjectured that this ratio is 3/2. On the other hand, he also mentions that there are examples where we cannot do better than T_∞, if both solutions are formed by the HDH algorithm in [82].

Seiden and Woeginger [141] observed that the APTAS of Kenyon and Rémila [90] can easily be adapted to produce a near-optimal packing in three stages for 2-D strip packing, showing that the asymptotic worst-case ratio of 2-D strip packing versus its k-stage version is 1 for any $k > 2$, and leading to an APTAS for the latter.

Bansal et al. [81] provided an APTAS for the *guillotine case*, i.e., the case in which the items have to be packed in alternate horizontal and vertical stages but there is no limit on the number of stages that can be used. In the guillotine case, there is a sequence of edge-to-edge cuts parallel to one of the edges of the bin. See Fig. 3(b) for an example of guillotine packing and Fig. 3(c) for an example that is not a guillotine packing. Recently Abed et al. [142] studied other related packing problems under guillotine cuts. They also made a conjecture that, for any set of n non-overlapping axis-parallel rectangles, there is a guillotine cutting sequence separating $\Omega(n)$ of them. A proof of this conjecture will imply a $O(1)$-approximation for Maximum Independent Set Rectangles, a related NP-hard problem. We refer the readers to [143] for more on guillotine packing.

3.7. Geometric knapsack

For 2-D GEOMETRIC KNAPSACK (GK), a result of Steinberg [76] for strip packing translates into a $(3+\varepsilon)$ approximation [144]. Present best known approximation algorithms for both the cases with and without orthogonal 90 degree rotations, are due to Jansen and Zhang [145] and has an approximation guarantee of $(2+\varepsilon)$. The same authors [146] also gave a faster and simpler $(2+\varepsilon)$-approximation for the unweighted case without rotations. On the other hand, only known inapproximability result is that there is no FPTAS unless P $=$ NP, even for packing squares into squares [111]. PTAS are known for special cases when resource augmentation is allowed in one dimension [147] for both the cases with and without rotations, all items are square [148,149] or all items are small [150]. Very recently, Heydrich and Wiese [149] gave an EPTAS for the square case. Bansal et al. [78] gave a PTAS for the special case when the range of the profit-to-area ratio of the rectangles is bounded by a constant for both the cases with and without rotations.

Recently Adamaszek and Wiese [151] gave a quasi-PTAS for weighted geometric knapsack for both the cases with and without rotations, assuming the input data to be quasi-polynomially bounded integers. Their techniques also extend the corridor and cycle decomposition techniques of [152] that states that there is a $(1+\varepsilon)$-approximate solution such that the knapsack is divided into $O_\varepsilon(1)$ thin corridors and each item is contained in one corridor. They partition the corridors into $(\log n)^{O_\varepsilon(1)}$ boxes and guessing of these boxes take $n^{(\log n)^{O_\varepsilon(1)}}$ time. Very recently, Abed et al. [142] obtained another quasi-PTAS for the version with guillotine cut under similar assumptions.

For 3-D, Diedrich et al. [153] have given $7+\varepsilon$ and $5+\varepsilon$ approximation, for the cases without and with rotations, respectively.

3.8. Other related problems

There are many other related geometric packing/covering problems such as MAXIMUM INDEPENDENT SET RECTANGLES [152], ONLINE SQUARE-INTO-SQUARE PACKING [154], GEOMETRIC SET COVER/HITTING SET PROBLEM [155], ANCHORED RECTANGLE PROBLEM [156], MAXIMAL AXIS-PARALLEL RECTANGLES [157], CIRCLE/SPHERE PACKING [158] etc. We do not cover these problems in this survey.

Table 3 summarizes present best results for strip packing and geometric knapsack. As previously, OFF denotes offline, ON denotes online, REC denotes rectangles, CUB denotes cubes, WR denotes with rotation and NR denotes without rotation.

4. Vector bin packing

In this section we survey the previous work on VECTOR PACKING and its variants.

Table 3
Present state of the art for strip packing and geometric knapsack.

Problem	Dim.	Subcase	Best algorithm	Best lower bound
Strip packing	2	OFF-REC-WR	asymp: PTAS [93]	NP-hard
			abs: $\frac{5}{3}+\varepsilon$ [121]	3/2
		OFF-REC-NR	asymp: PTAS [90]	NP-hard
	3	OFF-REC-NR	asymp: 1.5 [140]	$1+1/2196$ [94]
		OFF-CUB	asymp: PTAS [139]	NP-hard
	2	ON-REC-NR	asymp: T_∞ [130]	1.5401 [130]
	3	ON-REC-NR	asymp: 2.5545 [103][a]	1.907 [104]
	$d>3$	ON-REC-NR	asymp: T_∞^d [101]	$\to 3$ (for $d\to\infty$) [106]
Geometric knapsack	2	OFF-REC-NR	$2+\varepsilon$ [145]	No FPTAS [111]
		OFF-REC-WR	$2+\varepsilon$ [145]	No FPTAS [111]
		OFF-CUB	PTAS [148]	No FPTAS [111]
	3	OFF-REC-NR	$7+\varepsilon$ [153]	No FPTAS [111]
		OFF-REC-WR	$5+\varepsilon$ [153]	No FPTAS [111]

[a] See [106] for the modified algorithm.

4.1. Offline vector packing

The first paper to obtain an APTAS for 1-D bin packing by Fernandez de la Vega and Lueker [37], implies a $(d+\varepsilon)$ approximation for vector packing problem. Woeginger [159] showed that there exists no APTAS even for $d=2$ unless P = NP. However some restricted class of vectors may still admit an APTAS. For example, consider the usual partial order on d dimensional vectors, where $(x_1,x_2,\dots,x_d)\prec(y_1,y_2,\dots,y_d)$ if and only if $x_i\le y_i$ for all $i\in[d]$. The hard case for the lower bound comes when the items are pairwise incompatible. The opposite extreme case, when there is a total order on all items, is easy to approximate. In fact, a slight modification of de la Vega and Lueker [37] algorithm yields an APTAS for subproblems of d-dimensional VBP with constant Dilworth number. After nearly twenty years, offline results for the general case were improved by Chekuri and Khanna [160]. They gave an algorithm with asymptotic approximation ratio of $(1+\varepsilon d+H_{1/\varepsilon})$ where $H_k=1+1/2+\cdots+1/k$, is the k'th Harmonic number. Considering $\varepsilon=1/d$, they show that for fixed d, vector bin packing can be approximated to within $O(\ln d)$ in polynomial time. Bansal, Caprara and Sviridenko [86] then introduced the *Round and Approx* framework and the notion of subset oblivious algorithm and improved it further to $(1+\ln d)$. Both these algorithms run in time that is exponential in d (or worse). Yao [161] showed that no algorithm running in time $o(n\log n)$ can give better than a d-approximation.

For arbitrary d, Chekuri–Khanna [160] showed vector bin packing is hard to approximate to within a $d^{1/2-\varepsilon}$ factor for all fixed $\varepsilon>0$ using a reduction from GRAPH COLORING problem. This can be improved to $d^{1-\varepsilon}$ by using the following simple reduction. Let G be a graph on n vertices. In the d-dimensional VBP instance, there will be $d=n$ dimensions and n items, one for each vertex. For each vertex i, we create an item i that has size 1 in coordinate i and size $1/n$ in coordinate j for each neighbor j of i, and size 0 in every other coordinate. It is easily verified that a set of items S can be packed into a bin if and only if S is an independent set in G. Thus we mainly focus on the case when d is a fixed constant and not part of the input.

The two dimensional case has received special attention. Kellerer and Kotov [162] designed an algorithm for 2-D vector packing with worst case absolute approximation ratio as 2. On the other hand there is a hardness of 3/2 for absolute approximation ratio that comes from the hardness of 1-D bin packing.

Very recently, Bansal, Elias and Khan [88] have given improved approximation for multidimensional vector packing. They give a polynomial time algorithm with an asymptotic approximation guarantee of $(1+\ln(1.5)+\varepsilon)\approx(1.405+\varepsilon)$ for 2-D vector packing and a $\ln d+0.807+o_d(1)+\varepsilon$-approximation for d-dimensional vector packing. This overcomes a natural barrier of $(1+\ln d)$ of R&A framework due to the fact that one cannot obtain better than d-approximation using rounding based algorithms. They circumvent this problem based on two ideas.

First, they show a structural property of vector packing that any optimal packing of m bins can be transformed into nearly $\lceil\frac{3m}{2}\rceil$ bins of two types:

1. Either a bin contains at most two items and these items are large in at least one dimension, or
2. The bin has slack in one dimension (i.e., the sum of all vectors in the bin is at most $1-\delta$ for some constant δ). They then search (approximately) over the space of such "well-structured" 1.5-approximate solutions. However, as this structured solution (necessarily) uses unrounded item sizes, it is unclear how to search over the space of such solutions efficiently. So a key idea is to define this structure carefully based on matchings, and use an elegant recent algorithm for the multiobjective-multibudget matching problem by Chekuri, Vondrák, and Zenklusen [163].

The second step is to apply the subset oblivious framework to the above algorithm. There are two problems. First, the algorithm is not rounding-based. Second, even proving subset obliviousness for rounding based algorithms for vector packing is more involved than for geometric bin-packing. To get around these issues, they use additional technical observations about the structure of d-dimensional VBP.

Another consequence of these techniques is the following tight (absolute) approximation guarantee. They show that for any small constant $\varepsilon>0$, there is a polynomial time algorithm with an almost tight absolute approximation ratio of $(1.5+\varepsilon)$ for 2-D vector packing.

4.2. Online vector packing

A generalization of the FIRSTFIT algorithm by Garey et al. [164] gives $d+\frac{7}{10}$ competitive ratio for the online version. Galambos et al. [165] showed a lower bound on the performance ratio of online algorithms that tends to 2 as d grows. The gap persisted for a long time, and in fact it was conjectured in [166] that the lower bound is super constant, but sublinear. Azar et al. [167] settled the status by giving $\Omega(d^{1-\varepsilon})$ information theoretic lower bound using stochastic packing integer programs and ONLINE GRAPH COLORING problem [168] where a sequence of nodes arrives online along with edges to previously arrived neighbors and the *transparent* adversary reveals the color it gives to a node immediately after the online algorithm assigns its color. In fact their result holds for arbitrary bin size $B\in\mathbb{Z}^+$ if the bin is allowed to grow. In particular, they show that for any integer $B\ge1$, any deterministic online algorithm for VBP has a competitive ratio of $\Omega(d^{\frac{1}{B}-\varepsilon})$. For $\{0, 1\}$-VBP the lower bound is $\Omega(d^{\frac{1}{B+1}-\varepsilon})$. They also provided an improved upper bound for $B \geq 2$ with a polynomial time algorithm for the online VBP with competitive ratio: $O(d^{1/(B-1)} \log d^{B/(B+1)})$, for $[0, 1]^d$ vectors and $O(d^{1/B} \log d^{(B+1)/B})$, for $\{0, 1\}^d$ vectors. Recently, Azar et al. [169] studied the online vector packing for small vectors (relative to the size of a bin). For this special case when each vector's coordinates are at most $O(\varepsilon^2/\log d)$, they give a constant competitive ratio of $(1+\varepsilon)e$ for arbitrarily large d. For 2-D, they present a FirstFit variant with a competitive ratio ≈ 1.48 and another essentially tight algorithm (not via a FirstFit variant) with a competitive ratio arbitrarily close to 4/3. They also defined a *splittable model*, where a vector v can be split into arbitrary many fractions $v \cdot \alpha_1, v \cdot \alpha_2, \ldots, v \cdot \alpha_k$, $\sum_i \alpha_i = 1$ (here, each fractional vector $v \cdot \alpha_i$ can be placed into a different bin). In this setting, they gave an $e(1+\varepsilon)$-competitive algorithm. Later they [170] gave tight lower bound on the competitive ratio when vectors are small (even in the splittable, randomized setting) which approaches e for arbitrarily large d.

4.3. Vector knapsack

Knapsack (one-dimensional) problem is one of the most well-studied NP-complete problems in combinatorial optimization. There exists FPTAS by Ibarra and Kim [171], later on improved by Lawler [172]. We refer the readers to [173] for detailed survey on 1-D knapsack. However, for d-dimensional vector knapsack, it is well-known that there is no FPTAS unless P = NP, already for $d = 2$. Frieze and Clarke [79] gave a PTAS for d-dimensional knapsack using the dual simplex algorithm for LP. Subsequently, Caprara et al. [174] gave a scheme with improved running time of $O(n^{\lceil d/\varepsilon \rceil - d})$. However, there is no EPTAS even for 2-D vector knapsack [80], unless W[1] = FPT. In fact they used a reduction from a parameterized version of subset sum, known as sized subset sum, to show that unless all problems in SNP are solvable in sub-exponential time, there is no approximation scheme for 2-D vector knapsack whose running time is $f(1/\varepsilon)n^{o(\sqrt{1/\varepsilon})}$. Note that, for the case where $d = 1$, an EPTAS exists even for the multiple knapsack problem [175].

4.4. Vector scheduling

For $d = 1$, vector scheduling becomes the well-studied multiprocessor scheduling problem when jobs are scheduled into bins or *identical* machines. In a more general version, machine speeds can differ (called *uniform* machines) or jobs can have arbitrary different vector values in different machines (called *unrelated* machines). We refer the readers to the book on scheduling [176] for more details on different variants for scheduling problem. Vector scheduling (with identical machines) is strongly NP-hard, i.e., admits no FPTAS [177]. The classical result of Hochbaum and Shmoys [178] first gave a PTAS. A series of work [179,180] then culminated into an EPTAS with running time $O(2^{\tilde{O}(1/\varepsilon^2)} + n^{O(1)})$. The main idea of [180] is to use a fast IP in constant dimensions, together with existence of optimal integer solutions with small support using techniques from [181]. These results can mostly be extended to *uniform* machines [182,180].

For *d-dimensional offline* vector scheduling, the first major result was obtained by Chekuri and Khanna [160]. They obtained an algorithm with running time $n^{(1/\varepsilon)^{\tilde{O}(d)}}$, i.e., a PTAS when d is a fixed constant. PTASes are known for several other generalizations [183–185]. Bansal et al. [186] improved the running time of the PTAS to $O(2^{(1/\varepsilon)^{O(d \log\log(d))}} + nd)$. Using a reduction from 3-D matching they also showed that for any $\varepsilon < 1$, there is a $d(\varepsilon)$ so that there is no $(1+\varepsilon)$-approximation algorithm with running time $O(2^{(1/\varepsilon)^{o(d)}}(nd)^{O(1)})$, unless NP $\subseteq \cap_{\varepsilon>0}$ DTIME$(2^{n^{\varepsilon}})$. For arbitrary d, Chekuri and Khanna [160] obtained $O(\ln^2 d)$-approximation using approximation algorithms for packing integer programs (PIPs) as a subroutine. They also showed that, when m is the number of bins in the optimal solution, a simple random assignment gives $O(\ln dm/\ln\ln dm)$-approximation algorithm which works well when m is small. Furthermore, they showed that it is hard to approximate within any constant factor when d is arbitrary. This $\omega(1)$ lower bound is still the present best lower bound for the offline case. Harris and Srinivasan [187] gave a randomized $O(\log d/\log\log d)$-approximation algorithm using Moser–Tardos framework with partial resampling. In fact their algorithm works even for *unrelated machines*. We refer the readers to [188] for more related literature on vector scheduling.

In the *online setting*, for $d = 1$, Graham [189] gave a $(2 - 1/m)$-competitive algorithm for identical machines. Then a series of papers [190–194] led to the present best ratio of 1.9201 [195] and the present best lower bound of 1.880 [196] for deterministic algorithms. For unrelated machines, Aspnes et al. [197] obtained competitive ratio of $O(\log m)$ for makespan minimization. For arbitrary d and identical machines, Meyerson et al. [198] gave deterministic online algorithms with $O(\log d)$ competitive ratio. Im et al. [199] recently gave an algorithm with $O(\log d/\log\log d)$-competitive ratio. They also show tight information theoretic lower bound of $\Omega(\log d/\log\log d)$. Surprisingly this is also the present best offline algorithm! For unrelated machines, Meyerson et al. [198] gave $O(\log m + \log d)$-competitive algorithm. Im et al. [199] provided a matching lower bound. Note that we have used L_∞ norm in the definition of vector scheduling. However, other norms such as L_1 (for total machine load), L_2 (for disc storage) etc. have also been studied and Im et al. extends their results to asymptotic tight upper and lower bounds for any general L_p norm.

4.5. Vector bin covering

1-D vector bin covering was first investigated by Assman et al. [200,201]. They showed an online greedy algorithm is 2-competitive. Later Csirik and Totik [202] showed a tight information-theoretic lower bound to prove that the greedy algorithm achieves the best possible ratio. In fact from a reduction from partition problem, unless P = NP it is NP-hard to get $(2 - \varepsilon)$-approximation for offline bin covering. However, for the offline case, asymptotic worst case ratios of 3/2 and 4/3 was given [203]. Later, Csirik et al. [204] gave an APTAS for the problem. Finally, Jansen and Solis-Oba gave an AFPTAS [70] by using the potential price directive decomposition method of Grigoriadis and Khachiyan [205]. Csirik, Frenk, Galambos and Rinnooy Kan [206] gave a probabilistic analysis of 1-D and 2-D bin covering problems. Gaizer [207] gave an offline 2-approximation algorithm for $d = 2$. We refer the readers to [208] for some more related results. For d-dimensional vector bin covering problem, Alon et al. [209] gave an online algorithm with competitive ratio $2d$, for $d \geq 2$, and they showed an information theoretic lower bound of $\frac{2d+1}{2}$. For the offline version, they give an algorithm with an approximation guarantee of $O(\log d)$. For $d = 2$, they gave a simple 2-approximation algorithm and for $d \geq 2$, they used area of compact vector summation [210] to construct a simple d-approximation algorithm, which outperforms the $O(\log d)$-approximation algorithm for small d. Table 4 summarizes present best approximation/inapproximability results for vector packing and related variants.

5. Open problems

In this section we conclude by listing ten major open problems related to multidimensional bin packing.

Table 4
Present state of the art for vector packing and related variants.

Problem	Subcase	Best algorithm	Best lower bound
Vector bin packing	Offline (constant d)	$\ln d + 0.807 + o_d(1) + \varepsilon$ (asymp.[a]) [88]	APX-hard [159]
	Offline ($d = 2$)	$1.405 + \varepsilon$ (asymp.) [88]	APX-hard [159]
		$3/2 + \varepsilon$ (abs.[b]) [88]	3/2[c]
	Offline (arbitrary d)	$1 + \varepsilon d + O(\ln \frac{1}{\varepsilon})$(asymp.) [160]	$d^{1-\varepsilon}$[d]
	Online	$d + \frac{7}{10}$ [164]	$\Omega(d^{1-\varepsilon})$ [167]
Vector scheduling	Offline ($d = 1$)	EPTAS [180]	No FPTAS [177]
	Online ($d = 1$)	1.9201 [195]	1.880 [196]
	Offline (constant d)	PTAS [160]	No EPTAS [186]
	Offline (arbitrary d)	$O(\frac{\log d}{\log \log d})$ [187,199]	$\omega(1)$ [160]
	Online	$O(\frac{\log d}{\log \log d})$ [199]	$\Omega(\frac{\log d}{\log \log d})$ [199]
Vector knapsack	Offline ($d = 1$)	FPTAS [171,172]	NP-hard
	Offline (arbitrary d)	PTAS [174]	No EPTAS [80]
Vector bin covering	Offline ($d = 1$)	AFPTAS [70]	NP-hard
	Online ($d = 1$)	2 [201]	2 [202]
	Offline (arbitrary d)	$O(\log d)$ [199]	NP-hard
	Online (arbitrary d)	$2d$ [209]	$\frac{2d+1}{2}$ [209]

[a] Here asymp. means asymptotic approximation guarantee.
[b] Here abs. means absolute approximation guarantee.
[c] Follows from the fact that even 1-D bin packing cannot be approximated better than 3/2.
[d] See the reduction in Section 4.1.

Problem 1 (*Tight Approximability of Bin Packing*)**.** The present best polynomial-time algorithm for 1-D BP by Hoberg and Rothvoß [39], uses Opt $+ O(\log \text{Opt})$ bins. Proving one could compute a packing with only a constant number of extra bins will be a remarkable progress and is mentioned as one of the ten most important problems in approximation algorithms [30]. Consider the seemingly simple 3-Partition case in which all n items have sizes $s_i \in (1/4, 1/2)$. Recent progress by [50] suggests that either $O(\log n)$ bound is the best possible for 3-Partition or some fundamentally new ideas are needed to make progress.

Problem 2 (*Integrality Gap of Gilmore–Gomory LP*)**.** It has been conjectured in [47] that the Gilmore–Gomory LP for 1-D BP has *Modified Integer Roundup Property*, i.e., Opt $\le \lceil \text{Opt}_f \rceil + 1$. The conjecture has been proved true for the case when the instance contains at most 7 different item sizes [48]. Settling the status for the general case is an important open problem in optimization.

Problem 3 (*Tight Asymptotic Competitive Ratio for Online BP*)**.** The present best algorithm for online bin packing is by Heydrich and van Stee [59] who presented an online algorithm with asymptotic performance ratio of at most 1.5815 using a new type of *interval classification*. They also gave a lower bound of 1.5766 for any interval classification algorithm. In general the best known lower bound for asymptotic competitive ratio is 1.54014 [130]. Giving a stronger lower bound using some other construction is an important question in online algorithms. We also discussed $(A \otimes B)$-type algorithms for 2-D online GBP in Section 3.2 where two 1-D online bin packing algorithms A and B are used to give algorithm for 2-D online GBP. With the recent progress in 1-D online BP [59], it is an open question whether one can get $1.5815^2 \approx 2.502$ competitive ratio for 2-D online GBP.

Problem 4 (*Improved Approximability for Geometric Bin Packing and Strip Packing*)**.** There is a huge gap between the best approximation guarantee and hardness of geometric bin packing. There are no explicit inapproximability bounds known for multidimensional bin packing as function of d, apart from the APX-hardness in 2-D. Thus there is a huge gap between the best algorithm (1.69^{d-1}, i.e., exponential in d) and the hardness. Improved inapproximability, as a function of d, will be an interesting hardness result. One direction to get an improved approximation can be to extend R&A framework to d-D GBP or other related problems. However, there one key bottleneck is to find a good approximation algorithm to find the solution of the configuration LP. A $poly(d)$ asymptotic approximation for the LP will give us a $poly(d)$ asymptotic approximation for d-D GBP, a significant improvement over the current best ratio of $2^{O(d)}$ for $d > 2$.

Problem 5 (*Improved Approximability for Vector Bin Packing*)**.** Similarly, there are no explicit inapproximability bounds known for vector bin packing as function of d, apart from the APX-hardness in 2-D. Thus there is a gap between the best algorithm ($O(\ln d)$ for vector packing for $d > 2$) and the hardness. Improved inapproximability, as a function of d, will be an interesting hardness result even in this case.

Problem 6 (*Improved Approximability for Geometric Knapsack*)**.** Finding a PTAS for 2-D geometric knapsack (with or without rotations) is one of the major problems related to bin packing. Even for unweighted geometric knapsack (with or without rotations) factor 2 is the best known approximation.

Problem 7 (*Tight Ratio Between Optimal Guillotine Packing and Optimal Bin Packing*)**.** Improving the present guarantee for 2-D GBP will require an algorithm that is not *input-agnostic*. In particular, this implies that it should have the property that it can round two identical items (i.e., with identical height and width) differently. One such candidate is the guillotine packing approach [81]. It has been conjectured that this approach can give an approximation ratio of 4/3 for 2-D GBP. At present the best known upper bound on this gap is $T_\infty \approx 1.69$ [74]. Guillotine cutting also has connections with other geometric packing problems such as GEOMETRIC KNAPSACK and MAXIMUM INDEPENDENT SET RECTANGLES [142]. Though there is a QPTAS for guillotine knapsack (for inputs being quasi-polynomially bounded integers), there is no known polynomial time approximation. Finding a PTAS for 2-D guillotine knapsack is an interesting open problem.

Problem 8 (*Tight Ratio Between Optimal Two-Stage Packing and Optimal Bin Packing*)**.** Caprara conjectured [82] that there is a two-stage packing that gives 3/2 approximation for 2-D bin packing. As there are PTAS for 2-stage packing [74], this will give another 3/2 approximation for 2-D BP and coupled with R&A method this will give another $(1.405+\varepsilon)$ approximation. Presently the upper bound between best two-stage packing and optimal bin packing is $T_\infty \approx 1.69$. As 2-stage packings are very well-studied, this question is of independent interest and it might give us more insight on the power of Guillotine packing.

Problem 9 (*Tight Absolute Approximation for 2-D SP*)**.** As we had earlier mentioned, there is a gap between the best upper bound of $(5/3+\varepsilon)$ [121] and lower bound of 3/2. Tightening the gap is an interesting open problem.

Problem 10 (*Improved Pseudo-Polynomial Time Approximation*)**.** Pseudo-polynomial time (PPT) approximation is not well understood for geometric packing problems. In many cases we can break the barrier of polynomial time approximation by using pseudo-polynomial time approximation. For example, for strip packing there is PPT $\frac{4}{3}+\varepsilon$ approximation [123,124] whereas 3/2 is the polynomial time hardness (unless $\mathsf{P}=\mathsf{NP}$). Very recently, for strip packing a PPT hardness of 12/11 (unless $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{polylog(n)})$) was shown [125]. Finding tighter PPT hardness/algorithmic results for strip packing or other related problems will be interesting.

Finally, finding faster heuristics that work well in practice, is also a very important problem.

References

[1] Paul C. Gilmore, Ralph E. Gomory, A linear programming approach to the cutting-stock problem, Oper. Res. 9 (6) (1961) 849–859.
[2] Samuel Eilon, Nicos Christofides, The loading problem, Manage. Sci. 17 (5) (1971) 259–268.
[3] Richard W. Conway, William L. Maxwell, Louis W. Miller, Theory of Scheduling, 1967, Addison-Wesley, 1971.
[4] J.D. Ullman, The performance of a memory allocation algorithm. Technical Report 100, 1971.
[5] M.R. Garey, Ronald L. Graham, Jeffrey D. Ullman, Worst-case analysis of memory allocation algorithms, in: STOC, 1972, pp. 143–150.
[6] David S. Johnson, Near-optimal bin packing algorithms (Ph.D. thesis), Massachusetts Institute of Technology, USA, 1973.
[7] David S. Johnson, Alan Demers, Jeffrey D. Ullman, Michael R. Garey, Ronald L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, SIAM J. Comput. 3 (4) (1974) 299–325.
[8] Michael R. Garey, David S. Johnson, Computers and Interactibility: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., SF, CA, 1979.
[9] Dorit S. Hochbaum, Approximation Algorithms for NP-Hard Problems, PWS Publishing Co., 1996.
[10] David S. Johnson, Approximation algorithms for combinatorial problems, in: STOC, 1973, pp. 38–49.
[11] Teofilo F. Gonzalez, Handbook of Approximation Algorithms and Metaheuristics, CRC Press, 2007.
[12] Allan Borodin, Ran El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 2005.
[13] Steven Skiena, Who is interested in algorithms and why?: lessons from the stony brook algorithms repository, ACM SIGACT News 30 (3) (1999) 65–74.
[14] Michael R. Garey, David S. Johnson, Approximation algorithms for bin packing problems: A survey, in: Analysis and Design of Algorithms in Combinatorial Optimization, Springer, 1981, pp. 147–172.
[15] Edward G. Coffman Jr., Michael R. Garey, David S. Johnson, Approximation algorithms for bin-packin: an updated survey, in: Algorithm Design for Computer System Design, Springer, 1984, pp. 49–106.
[16] Edward G. Coffman, George S. Lueker, Probabilistic Analysis of Packing and Partitioning Algorithms, Wiley, New York, 1991.
[17] Gábor Galambos, Gerhard J. Woeginger, On-line bin packing - A restricted survey, Math. Methods OR 42 (1) (1995) 25–45.
[18] Edward G. Coffman, Gabor Galambos, Silvano Martello, Daniele Vigo, Bin packing approximation algorithms: Combinatorial analysis, in: Handbook of Combinatorial Optimization, Springer, 1999, pp. 151–207.
[19] János Csirik, Gerhard J. Woeginger, On-Line Packing and Covering Problems, Springer, 1998.
[20] Edward G. Coffman Jr., János Csirik, Gábor Galambos, Silvano Martello, Daniele Vigo, Bin packing approximation algorithms: Survey and classification, in: Handbook of Combinatorial Optimization, Springer, 2013, pp. 455–531.
[21] Balasubramanian Ram, The pallet loading problem: A survey, Int. J. Prod. Econ. 28 (2) (1992) 217–225.
[22] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, Udi Wieder, Heuristics for vector bin packing. research. microsoft. com, 2011.
[23] S.C. Sarin, W.E. Wilhelm, Prototype models for two-dimensional layout design of robot systems, IIE Trans. 16 (3) (1984) 206–215.
[24] Frits C.R. Spieksma, A branch-and-bound algorithm for the two-dimensional vector packing problem, Comput. Oper. Res. 21 (1) (1994) 19–25.
[25] Brenda S. Baker, Edward G. Coffman Jr., Ronald L. Rivest, Orthogonal packings in two dimensions, SIAM J. Comput. 9 (4) (1980) 846–855.
[26] Edward G. Coffman Jr., M.R. Garey, David S. Johnson, Robert Endre Tarjan, Performance bounds for level-oriented two-dimensional packing algorithms, SIAM J. Comput. 9 (4) (1980) 808–826.
[27] Andrea Lodi, Silvano Martello, Michele Monaci, Two-dimensional packing problems: A survey, European J. Oper. Res. 141 (2) (2002) 241–252.
[28] Narendra Karmarkar, Richard M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, in: FOCS, 1982, pp. 312–320.
[29] Vijay V. Vazirani, Approximation Algorithms, Springer Science & Business Media, 2001.
[30] David P. Williamson, David B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.
[31] Pierluigi Crescenzi, Alessandro Panconesi, Completeness in approximation classes, Inform. Comput. 93 (2) (1991) 241–262.
[32] Susanne Albers, Online algorithms: a survey, Math. Program. 97 (1–2) (2003) 3–26.
[33] Claire Kenyon, Best-fit bin-packing with random order, in: SODA, Vol. 96, 1996, pp. 359–364.
[34] Joan Boyar, Kim S. Larsen, Morten N. Nielsen, The accommodating function: A generalization of the competitive ratio, SIAM J. Comput. 31 (1) (2001) 233–258.
[35] Joan Boyar, Lene M. Favrholdt, The relative worst order ratio for online algorithms, ACM Trans. Algorithms (TALG) 3 (2) (2007) 22.
[36] Marc Demange, Pascal Grisoni, Vangelis Th Paschos, Differential approximation algorithms for some combinatorial optimization problems, Theoret. Comput. Sci. 209 (1) (1998) 107–122.
[37] Wenceslas Fernandez de la Vega, George S. Lueker, Bin packing can be solved within 1+epsilon in linear time, Combinatorica 1 (4) (1981) 349–355.
[38] Thomas Rothvoß, Approximating bin packing within o(log opt * log log opt) bins, in: FOCS, 2013, pp. 20–29.
[39] Rebecca Hoberg, Thomas Rothvoß, A logarithmic additive integrality gap for bin packing, in: SODA, 2017.
[40] Brenda S. Baker, Edward G. Coffman Jr., A tight asymptotic bound for next-fit-decreasing bin-packing, SIAM J. Algebr. Discrete Methods 2 (2) (1981) 147–152.
[41] György Dósa, Jiri Sgall, First fit bin packing: A tight analysis, in: STACS 2013, 2013, pp. 538–549.
[42] György Dósa, Rongheng Li, Xin Han, Zsolt Tuza, Tight absolute bound for first fit decreasing bin-packing: Ffd(l) $\leq$ 11/9 OPT(L) + 6/9, Theoret. Comput. Sci. 510 (2013) 13–61.
[43] György Dósa, Jirí Sgall, Optimal analysis of best fit bin packing, in: ICALP, 2014, pp. 429–441.
[44] Martin Grötschel, László Lovász, Alexander Schrijver, Geometric algorithm and combinatorial optimization, in: Algorithms and Combinatorics: Study and Research Texts, Vol. 2, Springer-Verlag, Berlin, 1988.
[45] S.A. Plotkin, D. Shmoys, E. Tardos, Fast approximate algorithm for fractional packing and covering problems, Math. Oper. Res. 20 (1995) 257–301.
[46] Sanjeev Arora, Elad Hazan, Satyen Kale, The multiplicative weights update method: a meta-algorithm and applications, Theory Comput. 8 (1) (2012) 121–164.
[47] Guntram Scheithauer, Johannes Terno, Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem, Oper. Res. Lett. 20 (2) (1997) 93–100.
[48] A. Sebö, G. Shmonin, Proof of the modified integer round-up conjecture for bin packing in dimension 7. Manuscript, 2009.
[49] Friedrich Eisenbrand, Dömötör Pálvölgyi, Thomas Rothvoß, Bin packing via discrepancy of permutations, ACM Trans. Algorithms 9 (3) (2013) 24.
[50] Alantha Newman, Ofer Neiman, Aleksandar Nikolov, Beck's three permutations conjecture: A counterexample and some consequences, in: FOCS, 2012, pp. 253–262.
[51] Thomas Rothvoß, The entropy rounding method in approximation algorithms, in: SODA, 2012, pp. 356–372.
[52] Shachar Lovett, Raghu Meka, Constructive discrepancy minimization by walking on the edges, in: FOCS, 2012, pp. 61–67.
[53] Michel X. Goemans, Thomas Rothvoß, Polynomiality for bin packing with a constant number of item types. in: SODA 2014, 2014, pp. 830–839.
[54] Klaus Jansen, Kim-Manuel Klein, About the structure of the integer cone and its application to bin packing, in: SODA, 2017, (in press).
[55] Klaus Jansen, Stefan Kratsch, Dániel Marx, Ildikó Schlotter, Bin packing with fixed number of bins revisited, J. Comput. System Sci. 79 (1) (2013) 39–49.
[56] C.C. Lee, D.T. Lee, A simple on-line bin-packing algorithm, J. ACM 32 (3) (1985) 562–572.
[57] Prakash Ramanan, Donna J. Brown, Chung-Chieh Lee, Der-Tsai Lee, On-line bin packing in linear time, J. Algorithms 10 (3) (1989) 305–326.
[58] Steven S. Seiden, On the online bin packing problem, J. ACM (JACM) 49 (5) (2002) 640–671.
[59] Sandy Heydrich, Rob van Stee, Beating the harmonic lower bound for online bin packing, in: ICALP, 2016, pp. 41:1–41:14.
[60] János Balogh, József Békési, Gábor Galambos, New lower bounds for certain classes of bin packing algorithms, Theoret. Comput. Sci. 440–441 (2012) 1–13.
[61] André van Vliet, An improved lower bound for on-line bin packing algorithms, Inform. Process. Lett. 43 (5) (1992) 277–284.
[62] János Balogh, József Békési, György Dósa, Jirí Sgall, Rob van Stee, The optimal absolute ratio for online bin packing, in: SODA, 2015, pp. 1425–1438.

[63] Rob van Stee, SIGACT news online algorithms column 20: the power of harmony, SIGACT News 43 (2) (2012) 127–136.
[64] Peter W. Shor, How to pack better than best fit: tight bounds for average-case online bin packing, in: FOCS, 1991, pp. 752–759.
[65] Janos Csirik, David S. Johnson, Claire Kenyon, James B. Orlin, Peter W. Shor, Richard R. Weber, On the sum-of-squares algorithm for bin packing, J. ACM (JACM) 53 (1) (2006) 1–65.
[66] Varun Gupta, Ana Radovanovic, Online stochastic bin packing, 2012. arXiv preprint arXiv:1211.2687.
[67] S. Tang, Q. Huang, X. Li, D. Wu, Smoothing the energy consumption: Peak demand reduction in smart grid, in: INFOCOM, IEEE, 2013, pp. 1133–1141.
[68] M. Karbasioun, G. Shaikhet, E. Kranakis, I. Lambadaris, Power strip packing of malleable demands in smart grid, in: IEEE International Conference on Communications, 2013, pp. 4261–4265.
[69] Anshu Ranjan, Pramod Khargonekar, Sartaj Sahni, Offline first fit scheduling in smart grids, in: IEEE SCC, 2015, pp. 758–763.
[70] Klaus Jansen, Roberto Solis-Oba, An asymptotic fully polynomial time approximation scheme for bin covering, Theoret. Comput. Sci. 306 (1) (2003) 543–551.
[71] Arindam Khan, Approximation algorithms for multidimensional bin packing (Ph.D. thesis), Georgia Institute of Technology, USA, 2015.
[72] Arindam Khan, Mohit Singh, On weighted bipartite edge coloring, in: FSTTCS, 2015, pp. 136–150.
[73] Aram Meir, Leo Moser, On packing of squares and cubes, J. Combin. Theory 5 (2) (1968) 126–134.
[74] Alberto Caprara, Andrea Lodi, Michele Monaci, Fast approximation schemes for two-stage, two-dimensional bin packing, Math. Oper. Res. 30 (1) (2005) 150–172.
[75] Nikhil Bansal, José R. Correa, Claire Kenyon, Maxim Sviridenko, Bin packing in multiple dimensions: Inapproximability results and approximation schemes, Math. Oper. Res. 31 (1) (2006) 31–49.
[76] A. Steinberg, A strip-packing algorithm with absolute performance bound 2, SIAM J. Comput. 26 (2) (1997) 401–409.
[77] Michael D. Grigoriadis, Leonid G. Khachiyan, Lorant Porkolab, J. Villavicencio, Approximate max–min resource sharing for structured concave optimization, SIAM J. Optim. 11 (4) (2001) 1081–1091.
[78] Nikhil Bansal, Alberto Caprara, Klaus Jansen, Lars Prädel, Maxim Sviridenko, A structural lemma in 2-dimensional packing, and its implications on approximability, in: ISAAC, 2009, pp. 77–86.
[79] A.M. Frieze, M.R.B. Clarke, Approximation algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses, European J. Oper. Res. 15 (1) (1984) 100–109.
[80] Ariel Kulik, Hadas Shachnai, There is no eptas for two-dimensional knapsack, Inform. Process. Lett. 110 (16) (2010) 707–710.
[81] Nikhil Bansal, Andrea Lodi, Maxim Sviridenko, A tale of two dimensional bin packing, in: FOCS,2005, pp. 657–666.
[82] Alberto Caprara, Packing 2-dimensional bins in harmony, in: FOCS, 2002, pp. 490–499.
[83] Klaus Jansen, Lars Prädel, New approximability results for two-dimensional bin packing, in: SODA, 2013.
[84] David R. Karger, Krzysztof Onak, Polynomial approximation schemes for smoothed and random instances of multidimensional packing problems, in: SODA, 2007, pp. 1207–1216.
[85] Leah Epstein, Harmonic algorithm for bin packing, in: Encyclopedia of Algorithms, 2015.
[86] Nikhil Bansal, Alberto Caprara, Maxim Sviridenko, A new approximation method for set covering problems, with applications to multidimensional bin packing, SIAM J. Comput. 39 (4) (2009) 1256–1278.
[87] Nikhil Bansal, Arindam Khan, Improved approximation algorithm for two-dimensional bin packing, in: SODA, 2014, pp. 13–25.
[88] Nikhil Bansal, Marek Eliás, Arindam Khan, Improved approximation for vector bin packing, in: SODA, 2016, pp. 1561–1579.
[89] F. Chung, M.R. Garey, David S. Johnson, On packing two-dimensional bins, SIAM J. Algebr. Discrete Methods 3 (1982) 66–76.
[90] Claire Kenyon, Eric Rémila, A near-optimal solution to a two-dimensional cutting stock problem, Math. Oper. Res. 25 (4) (2000) 645–656.
[91] Flavio Keidi Miyazawa, Yoshiko Wakabayashi, Packing problems with orthogonal rotations, in: LATIN, Springer, 2004, pp. 359–368.
[92] Leah Epstein, Rob van Stee, Optimal online algorithms for multidimensional packing problems, SIAM J. Comput. 35 (2) (2005) 431–448.
[93] Klaus Jansen, Rob van Stee, On strip packing with rotations, in: STOC, 2005, pp. 755–761.
[94] Miroslav Chlebík, Janka Chlebíková, Inapproximability results for orthogonal rectangle packing problems with rotations, in: CIAC, 2006, pp. 199–210.
[95] Guochuan Zhang, A 3-approximation algorithm for two-dimensional bin packing, Oper. Res. Lett. 33 (2) (2005) 121–126.
[96] Rolf Harren, Rob van Stee, Absolute approximation ratios for packing rectangles into bins, J. Sched. 15 (1) (2012) 63–75.
[97] Rolf Harren, Rob van Stee, Packing rectangles into 2opt bins using rotations, in: SWAT, 2008, pp. 306–318.
[98] Rolf Harren, Rob van Stee, Improved absolute approximation ratios for two-dimensional packing problems, in: APPROX-RANDOM, 2009, pp. 177–189.
[99] Klaus Jansen, Lars Prädel, Ulrich M. Schwarz, Two for one: Tight approximation of 2D bin packing, in; WADS, 2009, pp. 399–410.
[100] Don Coppersmith, Prabhakar Raghavan, Multidimensional on-line bin packing: algorithms and worst-case analysis, Oper. Res. Lett. 8 (1) (1989) 17–20.
[101] János Csirik, André Van Vliet, An on-line algorithm for multidimensional bin packing, Oper. Res. Lett. 13 (3) (1993) 149–158.
[102] Steven S. Seiden, Rob van Stee, New bounds for multi-dimensional packing, in: SODA, 2002, pp. 486–495.
[103] Xin Han, Francis Y.L. Chin, Hing-Fung Ting, Guochuan Zhang, Yong Zhang, A new upper bound 2.5545 on 2D online bin packing, ACM Trans. Algorithms 7 (4) (2011) 50.
[104] D. Blitz, A. van Vliet, G.J. Woeginger, Lower bounds on the asymptotic worst-case ratio of online bin packing alorithms. Manuscript, 1996.
[105] André Van Vliet, Lower and upper bounds for online bin packing and scheduling heuristics (Ph.D. thesis), Erasmus University, Netherlands, 1995.
[106] Rob van Stee, SIGACT news online algorithms column 26: Bin packing in multiple dimensions, ACM SIGACT News 46 (2) (2015) 105–112.
[107] Satoshi Fujita, Takeshi Hada, Two-dimensional on-line bin packing problem with rotatable items, Theoret. Comput. Sci. 289 (2) (2002) 939–952.
[108] Leah Epstein, Two-dimensional online bin packing with rotation, Theoret. Comput. Sci. 411 (31) (2010) 2899–2911.
[109] Leah Epstein, Two dimensional packing: the power of rotation, in: Mathematical Foundations of Computer Science 2003, Springer, 2003, pp. 398–407.
[110] Leah Epstein, Rob Van Stee, This side up!, in: ACM Transactions on Algorithms (TALG), Vol. 2, ACM, 2006, pp. 228–243.
[111] Joseph Y.T. Leung, Tommy W. Tam, Chin S. Wong, Gilbert H. Young, Francis Y.L. Chin, Packing squares into a square, J. Parallel Distrib. Comput. 10 (3) (1990) 271–275.
[112] Yoshiharu Kohayakawa, Flávio Keidi Miyazawa, Prabhakar Raghavan, Yoshiko Wakabayashi, Multidimensional cube packing, Electron. Notes Discrete Math. 7 (2001) 110–113.
[113] Leah Epstein, Rob van Stee, Online square and cube packing, Acta Inform. 41 (9) (2005) 595–606.
[114] Xin Han, Deshi Ye, Yong Zhou, A note on online hypercube packing, CEJOR Cent. Eur. J. Oper. Res. 18 (2) (2010) 221–239.
[115] Leah Epstein, Rob Van Stee, Bounds for online bounded space hypercube packing, Discrete Optim. 4 (2) (2007) 185–197.
[116] Sandy Heydrich, Rob van Stee, Improved lower bounds for online hypercube packing. CoRR, abs/1607.01229, 2016.
[117] Carlos E. Fereira, Flavio K. Miyazawa, Yoshiko Wakabayashi, Packing of squares into squares, in: Pesquisa Operacional, Citeseer, 1998.
[118] Nikhil Bansal, Maxim Sviridenko, Two-dimensional bin packing with one-dimensional resource augmentation, Discrete Optim. 4 (2) (2007) 143–153.
[119] Daniel Sleator, A 2.5 times optimal algorithm for packing in two dimensions, Inform. Process. Lett. 10 (1) (1980) 37–40.
[120] Ingo Schiermeyer, Reverse-fit: A 2-optimal algorithm for packing rectangles, in: ESA, Springer, 1994, pp. 290–299.
[121] Rolf Harren, Klaus Jansen, Lars Prädel, Rob van Stee, A $(5/3+\epsilon)$-approximation for strip packing, Comput. Geom. 47 (2) (2014) 248–267.
[122] Giorgi Nadiradze, Andreas Wiese, On approximating strip packing with a better ratio than 3/2, in: SODA, 2016, pp. 1491–1510.
[123] Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, Arindam Khan, Improved pseudo-polynomial-time approximation for strip packing, in: FSTTCS, 2016, pp. 9:1–9:14.
[124] Klaus Jansen, Malin Rau, Improved approximation for two dimensional strip packing with polynomial bounded width. CoRR, abs/1610.04430, 2016.
[125] Anna Adamaszek, Tomasz Kociumaka, Marcin Pilipczuk, Michal Pilipczuk, Hardness of approximation for strip packing. CoRR, abs/1610.07766, 2016.
[126] Igal Golan, Performance bounds for orthogonal oriented two-dimensional packing algorithms, SIAM J. Comput. 10 (3) (1981) 571–582.
[127] Brenda S. Baker, Donna J. Brown, Howard P. Katseff, A 5/4 algorithm for two-dimensional packing, J. Algorithms 2 (4) (1981) 348–368.
[128] Klaus Jansen, Roberto Solis-Oba, Rectangle packing with one-dimensional resource augmentation, Discrete Optim. 6 (3) (2009) 310–323.
[129] Brenda S. Baker, Jerald S. Schwarz, Shelf algorithms for two-dimensional packing problems, SIAM J. Comput. 12 (3) (1983) 508–525.
[130] János Csirik, Gerhard J. Woeginger, Shelf algorithms for on-line strip packing, Inform. Process. Lett. 63 (4) (1997) 171–175.
[131] Xin Han, Kazuo Iwama, Deshi Ye, Guochuan Zhang, Approximate strip packing: Revisited, Inform. Comput. (2016).
[132] Donna J. Brown, Brenda S. Baker, Howard P. Katseff, Lower bounds for on-line two-dimensional packing algorithms, Acta Inform. 18 (2) (1982) 207–225.
[133] Walter Kern, Jacob Jan Paulus, A tight analysis of brown-baker-katseff sequences for online strip packing, in: CTW, 2010, pp. 109–110.
[134] Rolf Harren, Walter Kern, Improved lower bound for online strip packing - (extended abstract), in: WAOA, 2011, pp. 211–218.
[135] Johann Hurink, Jacob Jan Paulus, Improved online algorithms for parallel job scheduling and strip packing, Theoret. Comput. Sci. 412 (7) (2011) 583–593.
[136] Deshi Ye, Xin Han, Guochuan Zhang, A note on online strip packing, J. Comb. Optim. 17 (4) (2009) 417–423.
[137] Li Keqin, Kam-Hoi Cheng, On three-dimensional packing, SIAM J. Comput. 19 (5) (1990) 847–867.
[138] Keqin Li, Kam-Hoi Cheng, Heuristic algorithms for on-line packing in three dimensions, J. Algorithms 13 (4) (1992) 589–605.
[139] Nikhil Bansal, Xin Han, Kazuo Iwama, Sviridenko Maxim, Guochuan Zhang, Harmonic algorithm for 3-dimensional strip packing problem, in: SODA, 2007, pp. 1197–1206.
[140] Klaus Jansen, Lars Prädel, A new asymptotic approximation algorithm for 3-dimensional strip packing, in: SOFSEM 2014: Theory and Practice of Computer Science, Springer, 2014, pp. 327–338.
[141] Steven S. Seiden, Gerhard J. Woeginger, The two-dimensional cutting stock problem revisited, Math. Program. 102 (3) (2005) 519–530.

[142] Fidaa Abed, Parinya Chalermsook, José R. Correa, Andreas Karrenbauer, Pablo Pérez-Lantero, José A. Soto, Andreas Wiese, On guillotine cutting sequences, in: APPROX, 2015, pp. 1–19.
[143] Enrico Pietrobuoni, Two-dimensional bin packing problem with guillotine restrictions (Ph.D. thesis), University of Bologna, Italy, 2015.
[144] Alberto Caprara, Michele Monaci, On the two-dimensional knapsack problem, Oper. Res. Lett. 32 (1) (2004) 5–14.
[145] Klaus Jansen, Guochuan Zhang, Maximizing the total profit of rectangles packed into a rectangle, Algorithmica 47 (3) (2007) 323–342.
[146] Klaus Jansen, Guochuan Zhang, Maximizing the number of packed rectangles, in: SWAT, 2004, pp. 362–371.
[147] Klaus Jansen, Roberto Solis-Oba, New approximability results for 2-dimensional packing problems, in: MFCS, 2007, pp. 103–114.
[148] Klaus Jansen, Roberto Solis-Oba, A polynomial time approximation scheme for the square packing problem, in: IPCO, 2008, pp. 184–198.
[149] Sandy Heydrich, Andreas Wiese, Faster approximation schemes for the two-dimensional knapsack problem, in: SODA, 2017, (in press).
[150] Aleksei V. Fishkin, Olga Gerber, Klaus Jansen, On efficient weighted rectangle packing with large resources, in: ISAAC, 2005, pp. 1039–1050.
[151] Anna Adamaszek, Andreas Wiese, A quasi-ptas for the two-dimensional geometric knapsack problem, in: SODA, 2015, pp. 1491–1505.
[152] Anna Adamaszek, Andreas Wiese, Approximation schemes for maximum weight independent set of rectangles, in: FOCS, 2013, pp. 400–409.
[153] Florian Diedrich, Rolf Harren, Klaus Jansen, Ralf Thöle, Henning Thomas, Approximation algorithms for 3D orthogonal knapsack, J. Comput. Sci. Technol. 23 (5) (2008) 749–762.
[154] Brian Brubach, Improved bound for online square-into-square packing, in: WAOA, 2014, pp. 47–58.
[155] Timothy M. Chan, Elyot Grant, Exact algorithms and apx-hardness results for geometric packing and covering problems, Comput. Geom. 47 (2) (2014) 112–124.
[156] Adrian Dumitrescu, Csaba D. Tóth, Packing anchored rectangles, Combinatorica 35 (1) (2015) 39–61.
[157] Adrian Dumitrescu, Minghui Jiang, Computational geometry column 60, SIGACT News 45 (4) (2014) 76–82.
[158] David de Laat, Frank Vallentin, A breakthrough in sphere packing: The search for magic functions. arXiv preprint arXiv:1607.02111, 2016.
[159] Gerhard J. Woeginger, There is no asymptotic ptas for two-dimensional vector packing, Inform. Process. Lett. 64 (6) (1997) 293–297.
[160] Chandra Chekuri, Sanjeev Khanna, On multidimensional packing problems, SIAM J. Comput. 33 (4) (2004) 837–851.
[161] Andrew Chi-Chih Yao, New algorithms for bin packing, J. ACM (JACM) 27 (2) (1980) 207–227.
[162] Hans Kellerer, Vladimir Kotov, An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing, Oper. Res. Lett. 31 (1) (2003) 35–41.
[163] Chandra Chekuri, Jan Vondrák, Rico Zenklusen, Multi-budgeted matchings and matroid intersection via dependent rounding, in: SODA, 2011, pp. 1080–1097.
[164] M.R. Garey, Ronald L. Graham, David S. Johnson, Resource constrained scheduling as generalized bin packing, J. Combin. Theory Ser. A 21 (3) (1976) 257–298.
[165] Gábor Galambos, Hans Kellerer, Gerhard J. Woeginger, A lower bound for on-line vector-packing algorithms, Acta Cybern. 11 (1–2) (1993) 23–34.
[166] Leah Epstein, On variable sized vector packing, Acta Cybern. 16 (1) (2003) 47–56.
[167] Yossi Azar, Ilan Reuven Cohen, Seny Kamara, Bruce Shepherd, Tight bounds for online vector bin packing, in: STOC, 2013, pp. 961–970.
[168] Magnús M. Halldórsson, Mario Szegedy, Lower bounds for on-line graph coloring, in: SODA, 1992, pp. 211–216.
[169] Yossi Azar, Ilan Reuven Cohen, Amos Fiat, Alan Roytman, Packing small vectors, in: SODA, 2016, pp. 1511–1525.
[170] Yossi Azar, Ilan Reuven Cohen, Alan Roytman, Online lower bounds via duality, in: SODA, 2017. (in press).
[171] Oscar H. Ibarra, Chul E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, J. ACM (JACM) 22 (4) (1975) 463–468.
[172] Eugene L. Lawler, Fast approximation algorithms for knapsack problems, Math. Oper. Res. 4 (4) (1979) 339–356.
[173] Hans Kellerer, Ulrich Pferschy, David Pisinger, Introduction to NP-Completeness of Knapsack Problems, Springer, 2004.
[174] Alberto Caprara, Hans Kellerer, Ulrich Pferschy, David Pisinger, Approximation algorithms for knapsack problems with cardinality constraints, European J. Oper. Res. 123 (2) (2000) 333–345.
[175] Klaus Jansen, Parameterized approximation scheme for the multiple knapsack problem, in: SODA, 2009, pp. 665–674.
[176] Michael Pinedo, Scheduling, Springer, 2015.
[177] Ulrich Faigle, Walter Kern, György Turán, On the performance of on-line algorithms for partition problems, Acta Cybernet. 9 (2) (1989) 107–119.
[178] Dorit S. Hochbaum, David B. Shmoys, Using dual approximation algorithms for scheduling problems theoretical and practical results, J. ACM (JACM) 34 (1) (1987) 144–162.
[179] Noga Alon, Yossi Azar, Gerhard J. Woeginger, Tal Yadid, Approximation schemes for scheduling on parallel machines, J. Sched. 1 (1) (1998) 55–66.
[180] Klaus Jansen, An eptas for scheduling jobs on uniform processors: using an milp relaxation with a constant number of integral variables, SIAM J. Discrete Math. 24 (2) (2010) 457–485.
[181] Friedrich Eisenbrand, Gennady Shmonin, Carathéodory bounds for integer cones, Oper. Res. Lett. 34 (5) (2006) 564–568.
[182] Dorit S. Hochbaum, David B. Shmoys, A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach, SIAM J. Comput. 17 (3) (1988) 539–551.
[183] Vincenzo Bonifaci, Andreas Wiese, Scheduling unrelated machines of few different types. CoRR, abs/1205.0974, 2012.
[184] Leah Epstein, Tamir Tassa, Vector assignment problems: a general framework, J. Algorithms 48 (2) (2003) 360–384.
[185] Leah Epstein, Tamir Tassa, Vector assignment schemes for asymmetric settings, Acta Inform. 42 (6–7) (2006) 501–514.
[186] Nikhil Bansal, Tim Oosterwijk, Tjark Vredeveld, Ruben van der Zwaan, Approximating vector scheduling: Almost matching upper and lower bounds, Algorithmica 76 (4) (2016) 1077–1096.
[187] David G. Harris, Aravind Srinivasan, The moser-tardos framework with partial resampling, in: FOCS, 2013, pp. 469–478.
[188] Tjark Vredeveld, Vector scheduling problems, in: Encyclopedia of Algorithms, 2016, pp. 2323–2326.
[189] Ronald L. Graham, Bounds for certain multiprocessing anomalies, Bell Syst. Tech. J. 45 (9) (1966) 1563–1581.
[190] Yair Bartal, Amos Fiat, Howard Karloff, Rakesh Vohra, New algorithms for an ancient scheduling problem, in: STOC, ACM, 1992, pp. 51–58.
[191] Yair Bartal, Howard Karloff, Yuval Rabani, A better lower bound for on-line scheduling, Inform. Process. Lett. 50 (3) (1994) 113–116.
[192] David R. Karger, Steven J. Phillips, Eric Torng, A better algorithm for an ancient scheduling problem, in: SODA, Vol. 94, 1994, pp. 132–140.
[193] Susanne Albers, Better bounds for online scheduling, SIAM J. Comput. 29 (2) (1999) 459–473.
[194] Todd Gormley, Nicholas Reingold, Eric Torng, Jeffery Westbrook, Generating adversaries for request-answer games, in: SODA, 2000.
[195] Rudolf Fleischer, Michaela Wahl, Online scheduling revisited, in: European Symposium on Algorithms, Springer, 2000, pp. 202–210.
[196] J.F. Rudin III., Improved bounds for the on-line scheduling problem (Ph.D. thesis), University of Texas at Dallas, USA, 2011.
[197] James Aspnes, Yossi Azar, Amos Fiat, Serge Plotkin, Orli Waarts, On-line routing of virtual circuits with applications to load balancing and machine scheduling, J. ACM (JACM) 44 (3) (1997) 486–504.
[198] Adam Meyerson, Alan Roytman, Brian Tagiku, Online multidimensional load balancing, in: APPROX, 2013, pp. 287–302.
[199] Sungjin Im, Nathaniel Kell, Janardhan Kulkarni, Debmalya Panigrahi, Tight bounds for online vector scheduling, in: FOCS, 2015, pp. 525–544.
[200] Susan Fera Assmann, Problems in discrete applied mathematics (Ph.D. thesis), MIT, USA, 1983.
[201] Susan Fera Assmann, David S. Johnson, Daniel J. Kleitman, JY-T. Leung, On a dual version of the one-dimensional bin packing problem, J. Algorithms 5 (4) (1984) 502–525.
[202] János Csirik, V. Totik, Online algorithms for a dual version of bin packing, Discrete Appl. Math. 21 (2) (1988) 163–167.
[203] János Csirik, J.B.G. Frenk, Shuzhong Zhang, Martine Labbé, Two simple algorithms for bin covering, Acta Cybernet. 14 (1) (1999) 13–25.
[204] Janos Csirik, David S. Johnson, Claire Kenyon, Better approximation algorithms for bin covering, in: SODA, 2001.
[205] Michael D. Grigoriadis, Leonid G. Khachiyan, Coordination complexity of parallel price-directive decomposition, Math. Oper. Res. 21 (2) (1996) 321–340.
[206] János Csirik, J.B.G. Frenk, Gábor Galambos, AHG Rinnooy Kan, Probabilistic analysis of algorithms for dual bin packing problems, J. Algorithms 12 (2) (1991) 189–203.
[207] T. Gaizer, An algorithm for the 2D dual bin packing problem. Unpublished manuscript, University of Szeged, Hungary, 1989.
[208] János Csirik, J.B.G. Frenk, A dual version of bin packing, Algorithms Rev. 1 (1990) 87–95.
[209] Noga Alon, Yossi Azar, János Csirik, Leah Epstein, Sergey V. Sevastianov, Arjen P.A. Vestjens, Gerhard J. Woeginger, On-line and off-line approximation algorithms for vector covering problems, Algorithmica 21 (1) (1998) 104–118.
[210] Sergey Vasil'evich Sevast'janov, on some geometric methods in scheduling theory: A survey, Discrete Appl. Math. 55 (1) (1994) 59–82.