São Paulo Journal of Mathematical Sciences (2022) 16:585–615
https://doi.org/10.1007/s40863-022-00301-3

SPECIAL ISSUE COMMEMORATING THE GOLDEN JUBILEE OF THE INSTITUTE OF MATHEMATICS AND STATISTICS OF THE UNIVERSITY OF SÃO PAULO

Techniques and results on approximation algorithms for packing circles

Flávio K. Miyazawa[1] · **Yoshiko Wakabayashi**[2]

Accepted: 9 March 2022 / Published online: 11 April 2022

Abstract
This survey provides an introductory guide to some techniques used in the design of approximation algorithms for circle packing problems. We address three such packing problems, in which the circles may have different sizes. They differ on the type of the recipient. We consider the classical bin packing and strip packing, and a variant called knapsack packing. Our aim is to discuss some techniques and basic algorithms to motivate the reader to investigate these and other related problems. We also present the ideas used on more elaborated algorithms, without going into details, and mention known results on these problems.

Keywords Circle packing · Approximation · Bin packing · Strip packing · Knapsack

Mathematics Subject Classification 68W25 · 68W27 · 68Q17 · 52C17 · 52C26

1 Introduction

Problems on packing of objects like circles and spheres have always attracted great attention of mathematicians and have been investigated for many centuries.

Communicated by Yoshiharu Kohayakawa.

Dedicated to the memory of Imre Simon (1943–2009), who continues to inspire us.

✉ Flávio K. Miyazawa
fkm@ic.unicamp.br

Yoshiko Wakabayashi
yw@ime.usp.br

1 Institute of Computing, University of Campinas, Campinas, Brazil

2 Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil

One of the most famous packing problems concerns a question on the densest possible arrangement of equally sized spheres in the three-dimensional Euclidean space. A solution to this problem was conjectured by Johannes Kepler [38] in 1611, but only in 1998 Thomas Hales [31] announced to have found a proof. Hales's proof is based on exhaustive tests making use of complex computer calculations. Several years later, in 2017, a formal proof presented by Hales et al. [32] was published in the journal *Forum of Mathematics, Pi*.

The corresponding highest-density arrangement of equally sized circles on the Euclidean plane was studied by Joseph Louis Lagrange (in 1773) and also by Carl Friedrich Gauß (in 1831) and proofs were presented by Axel Thue [52] (in 1910) and László Fejes-Tóth [20] (in 1940) that the hexagonal lattice is the asymptotically densest of all possible circle packings (yielding a density close to $\pi/\sqrt{12} \approx 0.9069$). For results on packing of identical circles, we refer the reader to [3, 5, 19, 26, 28, 51].

While these highest-density arrangement/packing problems were object of study of mathematicians since many centuries ago, more recently, other types of packing problems have been considered in the literature. Since 1970, they have attracted more attention of computer scientists, interested both in theoretical issues concerning their computational complexity, as well as more practical issues concerning the design of algorithms to solve them.

Most classical packing problems are NP-hard, in particular, the problems we shall focus here. Such negative results lead naturally to question whether it is possible to obtain efficient algorithms for these problems that provide solutions that are reasonably close to the optimal ones.

Of course, to answer this question appropriately, we have to define formally what we mean by "efficient" algorithms and "reasonably close" solutions. Roughly speaking, we can say that the answer is yes for many packing problems. In fact, the algorithms we shall consider here—called approximation algorithms—are of this nature. They are required to be polynomial (in the size of the instance) and must provide a solution with the guarantee that its value is within some factor of the optimum value, for any instance of the problem. Moreover, such a guarantee has to be shown with formal proofs, and should not be based on experimental results.

The concept of approximation algorithms will be defined formally in the next subsection. For the moment, we want to mention that for some NP-hard problems no approximation algorithm exists, and for others, they may exist, but with not so good approximation factor.

Approximation algorithms for NP-hard problems were investigated even before the proof of the existence of NP-complete problems [16, 29], and Garey, Graham, and Ullman [27], as well as Johnson [37], formalized the concept of approximation algorithms. Johnson's PhD thesis [36] on approximation algorithms for the bin packing problem contains beautiful and ingenious ideas to analyse relatively simple algorithms that were proposed for the one-dimensional case. These pioneering works opened up an area of research that has become very central in the design of algorithms. For surveys on approximation algorithms for packing problems, see [7, 9, 10, 43].

In this work, we focus on packing problems where items (geometric objects) of a certain type (circles, squares, spheres) must be packed inside recipients of predefined types. A packing is a non-overlapping placement of a given set of items into one or more recipients in such a way that every item is fully contained in the recipient(s). The objective may be, for example, to minimize the number of recipients (of fixed size) needed to pack a given set of items, or minimize the height of a rectangular strip of fixed width (in the two-dimensional case) or to maximize the density of the packing. When the items are circles, these problems are known as *circle bin packing problem*, *circle strip packing problem* and *circle knapsack problem.*

We hope this material will be useful for those who are not familiar with this topic and also for those who have worked on packing problems. We present first some basic techniques, and then we address a more elaborate approach that has been used more recently, discussing the main ideas, without going into much detail.

2 Notation and preliminaries

Given a real-valued function f defined on a discrete set S, we denote by $f(S)$ the value $\sum_{i\in S} f(i)$. The *diameter* of a geometric object X, denoted by diam(X), is the largest distance between any two points of the object. Given two objects X and Y, we say that X is *small compared to* Y if $\mathrm{diam}(X) = O(\varepsilon)\mathrm{diam}(Y)$ for a small constant $\varepsilon > 0$.

Throughout this text, an input set of circles is given by a list of indices $L = \{1, \dots, n\}$, where each $i \in L$ corresponds to a circle with radius r_i and area a_i. For a subset $S \subseteq L$, the area of S, denoted by Area (S), is the sum of the areas of the elements in S. We also denote by Area (ρ) the area of a circle of radius ρ. The recipients, called *bins*, are usually unit squares or rectangular strips of width 1. Without loss of generality, we consider that $r_i \le 1/2$.

A *packing* of a set of circles L into a two-dimensional (2D) bin B is a function that maps the center of each circle $i \in L$ into the region that defines B, satisfying the following properties: each packed circle must be totally contained within the region of B, and the inner region of any two distinct packed circles may not intersect.

The problems to be addressed here can be formally defined as follows:

Circle bin packing (CBP): Given a list $\mathcal{C}$ of circles, find a packing of $\mathcal{C}$ into a minimum number of unit square bins.

Circle strip packing (CSP): Given a list $\mathcal{C}$ of circles, find a packing of $\mathcal{C}$ into a rectangular strip of width 1 so as to minimize the height of the rectangular area that contains the packed circles.

Circle knapsack (CK): Given a list $\mathcal{C}$ of circles and values $v_i > 0$ for each $i \in \mathcal{C}$, find a subset $\mathcal{S} \subseteq \mathcal{C}$ and a packing of $\mathcal{S}$ into a unit square bin such that $v(\mathcal{S})$ is maximized.

Sometimes, we will use bins of different sizes. In this case, if a bin has width w and height h, we denote an instance for the CBP and CK problems by a tuple (L, w, h); and if the strip has width w, we denote by (L, w) an instance of the CSP problem. In any case, we consider that w and h are constants.

The corresponding problems for packing spheres are defined analogously. We refer to them as 3D versions of the previous ones. They are called SPHERE BIN PACKING (SBP), SPHERE STRIP PACKING (SSP), and SPHERE KNAPSACK (SK) problems.

The problems we have mentioned are known to be computationally very complex. Additionally, they raise a difficult issue concerning representation of their solutions. Later, we shall discuss these issues and possible solutions to deal with them.

In 2010, Demaine, Fekete and Lang [12] proved that the problem of deciding whether a set of circles can be packed into a unit square (or into an equilateral triangle) is NP-hard. This implies that the 2D versions of all circle packing problems we mentioned above are NP-hard, and therefore, unless P = NP, there is no polynomial-time algorithm for any of them. Although this does not directly translate to the corresponding hardness of their 3D versions, most likely they are NP-hard as well.

In view of these negative results, the design of approximation algorithms is highly welcome, especially because packing problems have many applications, and efficient algorithms are necessary to deal with medium to large instances. These algorithms are defined formally as follows.

Given an algorithm $\mathcal{A}$ for a problem Π, and an instance I for Π, we denote by $\mathcal{A}(I)$ the value of the solution produced by $\mathcal{A}$, and by $\mathrm{OPT}_{\Pi}(I)$ the value of an optimal solution for I. For some $\alpha \geq 1$, we say that a polynomial-time algorithm $\mathcal{A}$, for a minimization problem, is an *asymptotic α-approximation algorithm* if, for every instance I, we have $\mathcal{A}(I) \leq \alpha\,\mathrm{OPT}_{\Pi}(I) + O(1)$. We also say that $\mathcal{A}$ is an asymptotic approximation algorithm with *factor* (or ratio) α. A family of polynomial-time algorithms $\{\mathcal{A}_{\varepsilon}\}$, for any fixed $\varepsilon > 0$, is said to be an *asymptotic polynomial-time approximation scheme* (APTAS) for a problem Π if, for every instance I, we have $\mathcal{A}_{\varepsilon}(I) \leq (1+\varepsilon)\,\mathrm{OPT}_{\Pi}(I) + O(1)$. If the constant term $O(1)$ is omitted from the definitions, then we say that $\mathcal{A}$ is an *α-approximation algorithm*, and $\{\mathcal{A}_{\varepsilon}\}$ is a *polynomial-time approximation scheme* (PTAS), respectively. In all cases, the approximation factor α need not be a constant: it can be a function that depends on I. The definitions for maximization problems are analogous (we reverse the inequality sign and take $\alpha \leq 1$ or $(1-\varepsilon)$). When the problem Π is clear from the context, we simply write OPT.

When dealing with packing problems, one is mostly interested in basically two types of algorithms: offline or online. An *offline algorithm* may use information on all items to be packed before finding a packing. An *online algorithm* receives the items one after the other, and must pack each item immediately into a bin (at the time it appears), without the knowledge of items that will appear in the future (the entire input is not available from the start). Moreover, once an item is packed into a bin, it cannot be removed from the position where it is packed. There is also the notion of *semi-online* algorithms (see [10]), but the existing approximation algorithms for packing circles focus on offline and online algorithms.

A difficulty that arises when dealing with circle packing problems concerns the representation of their solutions (the specification of the position in the bin in which each item must be packed). Typically, to measure the computational complexity of a problem, the time complexity is given in terms of the size of the input instance (number of bits used to represent it), which can be handled if the numbers that appear in the input and output are rational. But for the circle bin packing problem, for instance, it is not clear whether every instance, with bins and items given by rational numbers, always has an optimal solution with center coordinates that are also rational numbers. To deal with this problem, we may consider augmented bins (of a very small factor), and now, any packing (with possible irrational coordinates) can be rearranged to one using rational positions in an augmented bin. Although the algorithms use augmented bins, we compare the obtained solutions with the value of optimum packings using unit bins.

As in the CSP and SSP problems we are concerned with packings into one bin of unlimited height, and we want to minimize the height of the packing, there is no need to consider an augmented bin.

Many algorithms for packing problems subdivide the input list into sublists and generate specific packings for each sublist, and finally concatenate these packings to obtain a final packing of the input list. Given two packings $\mathcal{P}_1$ and $\mathcal{P}_2$ for a packing problem Π, we denote by $\mathcal{P}_1 \| \mathcal{P}_2$ the concatenation of these two packings. For the CBP problem, the concatenation is a union of the packed bins of the two packings, and for the CSP problem the packing $\mathcal{P}_1 \| \mathcal{P}_2$ is a packing containing packings $\mathcal{P}_1$ and $\mathcal{P}_2$ within the same rectangular strip, but with $\mathcal{P}_2$ starting at the top of $\mathcal{P}_1$.

When we design approximation algorithms, it is important to consider how we can compare, for each instance I, the value of a solution returned by the algorithm with the value of an optimal solution. For that, we may use some argument (based on a perfect packing, even if it is not feasible), or we may exhibit another structure (a feasible solution) that can be computed efficiently, whose objective value is a bound for the value of an optimal solution.

An example of a simple argument, in the two-dimensional case, is one based on the area of the items in the input list. As we are considering bins where the bounded edge size is equal to 1, the following lemma is straightforward, and will be useful in the forthcoming analyses of some area-based algorithms.

Lemma 1 *Let Π be one of the problems* CSP *or* CBP. *Then*, *for any instance I for the problem Π*, *we have* $\text{Area}\,(I) \leq \text{OPT}_{\Pi}(I)$.

After presenting some simple area-based algorithms, we discuss why area-based arguments may not give very good asymptotic approximation factors.

To obtain an APTAS, one has to use more elaborated ideas. In 2016, Miyazawa, Pedrosa, Schoeury, Sviridenko and Wakabayashi [48] presented an APTAS for the circle bin packing problem with augmented bins and an APTAS for the strip packing problem. In the special case of the circle knapsack problem in which the value of each circle is its area, Lintzmayer, Miyazawa and Xavier [41] obtained an APTAS, when an augmented bin can be used. They are based on a technique we call here *gap-structured partition*, which is central to these algorithms. We discuss the ideas behind this technique.

3 Circle bin packing

Most of the approximation algorithms for packing problems are asymptotic approximation algorithms. One of the reasons is because for many of these problems approximation algorithms with good factors do not exist, under the hypothesis that P $\neq$ NP. For example, it is not difficult to prove that (under this hypothesis), there is no approximation algorithm with factor smaller than 2 for the CBP problem. To see this, suppose there exists a polynomial-time algorithm $\mathcal{A}$ for CBP with approximation factor $\alpha < 2$. We can use $\mathcal{A}$ to decide in polynomial time whether a list of circles L can be packed into a (unit) bin. If the circles in L can be packed into a bin, algorithm $\mathcal{A}$ applied on L cannot use 2 bins, as $\alpha < 2$, and so it must produce a packing of L into one bin. Thus, we can decide in polynomial time if a list of circles can be packed into a bin, a problem known to be NP-hard [12].

On the other hand, asymptotic approximation algorithms with better ratios can be obtained. As we will see, for most of the problems mentioned here there are asymptotic approximation schemes. In what follows, we will concentrate on the asympotic case.

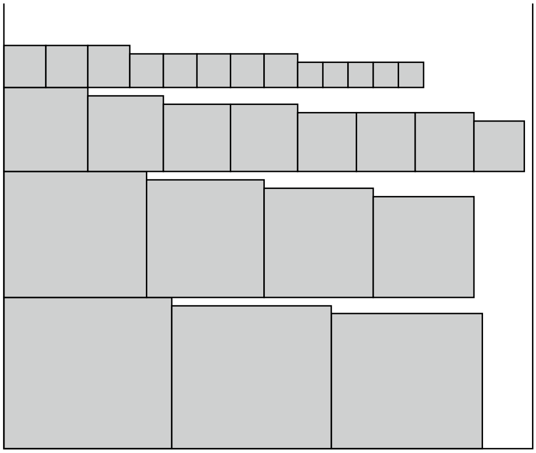
3.1 Area-based algorithms

As we mentioned, the area of the objects in the input list is a lower bound for an optimum value. Thus, it is natural to consider algorithms that can take advantage of this fact, guaranteeing a minimum area occupation in each bin, except perhaps for a few of them.

A straightforward idea for this problem is to round circles to shapes for which known packing algorithms exist. For example, in the literature we find many algorithms to pack squares; thus one possible idea is to inscribe each circle in a square and use these algorithms to obtain packing for circles. This can be done in the d-dimensional case, if algorithms for d-dimensional cubes are available. Given an algorithm $\mathcal{A}$ for a d-dimensional cube packing problem, we denote by $\mathcal{A}^{\circ}$ the corresponding algorithm that encloses each d-dimensional sphere into a d-dimensional cube and uses the packing algorithm $\mathcal{A}$ to solve the problem.

We exemplify this idea in the 2-dimensional case. For that, let us consider a well-known packing algorithm called NFDH (Next Fit Decreasing Height) [11, 47], designed for the rectangular strip and bin packing problems. To pack squares, this algorithm works as follows: it packs the input items following a non-increasing order of their sizes. The items are packed over shelves (of zero thickness) side by side from left to right, starting at the bottom of the bin. As items are sorted in non-increasing order of their sizes, the height of each shelf is defined by the first item it receives. When an item cannot be packed into a shelf, it creates a new shelf over the previous shelf, before placing that item. For the rectangular bin packing version, if a shelf cannot be created in the current bin, then it is created in the bottom of a new bin (see Fig. 1).

Fig. 1 Example of a packing generated by the NFDH algorithm



Modifying the previous algorithm, by changing the idea of "Next Fit" to "First Fit", we obtain an algorithm called FFDH (First Fit Decreasing Height). This algorithm first tries to pack the current item in the end of the first existing shelf with room to accommodate it; and, if necessary, creates a new shelf (with the same height of the item) at the top of the current packing. The description of these algorithms are for the strip packing problem, but can be converted to the bin packing problem using a one-dimensional packing algorithm to pack shelves into bins. In particular, the algorithm called HFF (Hybrid First Fit) uses FFDH to generate shelves and uses again a first fit approach to pack shelves into two dimensional bins. All these algorithms are offline and were designed for the d-dimensional case; for more details, see [8].

When the items to be packed are squares, each one with side length at most $1/m$, for an integer $m \geq 1$, it is known that HFF guarantees an area occupation factor in each bin of at least $(\frac{m}{m+1})^2$, except perhaps for the last bin. As the process of inscribing circles in squares increases the area by a factor of $4/\pi \approx 1.274$, the following result holds.

Lemma 2 *Let L be a list of circles, each one with diameter at most $1/m$, for an integer $m \geq 1$. Then,* $\mathrm{HFF}^{\circ}(L) \leq \frac{4}{\pi}\left(\frac{m+1}{m}\right)^2 \mathrm{Area}\,(L) + 1$.

For the general case of CBP problem, that is, when $m = 1$, the above result combined with Lemma 1 gives us that HFF° has an asymptotic approximation factor that is at most $4 \cdot 1.274 \approx 5.093$.

Proposition 1 *For any list L of circles for the* CBP *problem, we have that* $\mathrm{HFF}^{\circ}(L) \leq 5.093\ \mathrm{OPT}_{\mathrm{CBP}}(L) + 1$.

Recently, Fekete et al. [24] showed an algorithm called Split Packing, which we denote here by $\mathcal{A}_{\mathrm{SP}}$, that packs any set of circles into a square bin if its total area is at most a factor of $\gamma = 0.5390\ldots$ of the square bin. This factor is tight and best possible, and is called *critical density* [24]. Using $\mathcal{A}_{\text{SP}}$, we can design a simple algorithm for the CBP problem that has an asymptotic approximation factor at most $2/\gamma$. Let us call it, $\mathcal{A}_{\text{SP}}^{\text{CBP}}$. It proceed as follows:

Sort the items in L in decreasing order of their sizes. Partition the input list L into sublists $L_1, L_2, \ldots$, obtained by inserting each circle $c \in L$ in the first list L_i if $\text{Area}\,(L_i) + \text{Area}\,(c)$ is at most γ. If there is no such list, insert c in a new empty list L_{i+1}.

Clearly, at the end of this procedure, we have that each list L_i has only one circle or its total area is at most γ. Moreover, each sublist has total area at least $\gamma/2$, except perhaps the last one. If a sublist has only one circle, we can pack it in the center of the bin, while if the total area is at most γ, we can use $\mathcal{A}_{\text{SP}}$.

Denote by N the number of bins used by $\mathcal{A}_{\text{SP}}^{\text{CBP}}$. Consider $N \geq 2$, otherwise the result is immediate. Since each bin, except perhaps the last one, has an area occupation of at least $\gamma/2$, we have $\text{Area}\,(L) \geq (N-1)\frac{\gamma}{2}$, and therefore,

$$\mathcal{A}_{\text{SP}}^{\text{CBP}}(L) \leq \frac{2}{\gamma}\,\text{Area}\,(L) + 1 \leq 3.7105\,\text{OPT}_{\text{CBP}}(L) + 1.$$

If we only consider packing arguments based on the straighforward use of area, we cannot expect a better factor. Suppose all circles in the input list have radius $\rho_2 + \varepsilon$, where $\varepsilon > 0$ is a small value and $\rho_2 = \frac{1}{2+\sqrt{2}}$. The value of ρ_2 is the largest radius of two equal circles that can be packed in a unit square bin. The area of each circle of radius $\rho_2 + \varepsilon$ can be made as close to $\gamma/2$ as desired, using a sufficiently small value of ε.

In general, for small circles we can obtain better area occupation. For example, using the algorithm stated in Lemma 2, the area occupation in each bin tends to $\pi/4 \approx 0.7854$ (and for very small equally sized circles we can produce hexagonal packings with area occupation that tends to $\pi/\sqrt{12} \approx 0.9069$ as their sizes become smaller). These ideas would lead to better approximation algorithms when circles become smaller. But using only area occupation argument, it is not possible to have an analysis that will lead to asymptotic approximation schemes, as we may have input lists with small circles of the same size and the hexagonal packing is the best one can do. To obtain better approximation algorithms, there is a need to analyse more closely the structure of optimum packings.

3.2 Packing small circles

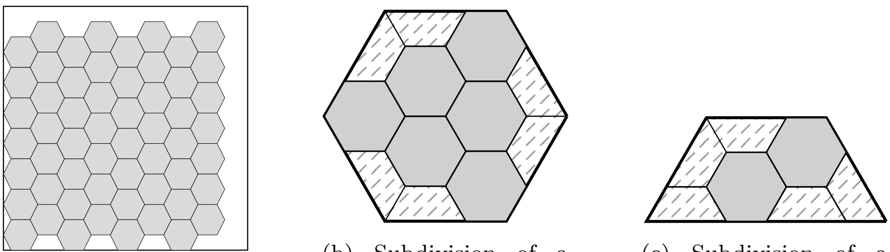
Before presenting the APTAS for the circle bin packing problem, we present a way to pack small circles guaranteeing an area occupation close to $\pi/\sqrt{12}$ (the maximum possible circle density, as the circles become smaller). Note that, as the squares become smaller, the NFDH algorithm, mentioned in Sect. 3.1, obtains an area occupation that approaches 1. On the other hand, if we inscribe each circle in a square, we cannot guarantee the mentioned area occupation anymore. A possibility would be to subdivide the input list of small circles into sublists with similar radii and obtain a hexagonal packing shape for each sublist. However, we may not subdivide the input list in too many sublists, as this would increase the additive term β, that appears in the inequality of the asymptotic approximation bound, to a non-constant value.

To tackle this problem, Hokama et al. [34] presented an algorithm that combines the subdivision of the input list with the subdivisions of the bins. Their idea is to consider a recursive subdivision of hexagonal or trapezoidal (half hexagons) bins into smaller sub-bins of hexagonal or trapezoidal shapes, as in Figs. 2b and 2c. In this subdivision, a hexagon of side length ℓ can be partitioned into six smaller hexagons of side length $\ell/3$ and other six trapezoids, each one with three sides of length $\ell/3$ and one side with length $2\ell/3$. The idea is to subdivide the small circles into K types based on their radii and pack them into hexagonal bins of the same type. For each type, the first hexagonal bins are obtained by the hexagonal tiling of a unit square bin, as in Fig. 2a. Moreover, circles and bins of each type are classified by subtypes, also based on their radii, so that a circle of one subtype is packed in a hexagonal bin obtained by sudivisions of hexagonal or trapezoidal bins of larger subtype. The authors show that the average area loss in each bin, given by the area of the bins subtracted by the total area of the circles, is due to the hexagonal tiling of the unit square bins and the packing of circles into hexagons. They prove that an area occupation in each bin can be made, asymptotically, close to $\pi/\sqrt{12}$, as the circles become smaller and the number of types increases.

3.3 Using (quasi-) optimal subpackings

Although larger circles lead to worse area occupation guarantee, it is easier to obtain optimum packings for them. So, a first idea to take advantage of this fact is to subdivide the input list into a sublist of large circles and a sublist of small circles and obtain a combined packing, consisting of an optimum packing for the first sublist and a packing with good area occupation for the second sublist.

To use this idea, we may consider the problem of packing k largest equally sized circles into a unit square, for which there is an extensive literature [33, 44, 51]. We denote by ρ_k the largest radius for which it is possible to pack k circles with radius ρ_k into a unit square. Optimum configuration C_k to pack k such circles is known for many values of k. The website maintained by Specht [50] presents



(a) Hexagonal Packing (b) Subdivision of a hexagonal bin (c) Subdivision of a trapezoidal bin

Fig. 2 Recursive subdivision of hexagonal/trapezoidal bins into smaller hexagonal or trapezoidal shapes references and the best known values for ρ_k, for k up to 30 [18, 45]. Recently, values of ρ_k, for $k \in \{31, 32, 33\}$, have been obtained by Markót [46]. If we denote by L_k the circles in the input list with radii in the interval $(\rho_{k+1}, \rho_k]$, an algorithm to pack L_k can be obtained by rounding up each circle of L_k to have radius ρ_k and then using packing configurations C_k, for each bin, except perhaps for the last bin, which may have fewer circles.

Based on the results that tell us the optimum values of ρ_k, we can obtain the least area occupation if we pack k circles that belong to the list L_k. For $k \leq 8$, these values are shown in Table 1.

In Table 1, we can see that the worst area occupation is given by the (large) circles in list L_1 that has an area occupation of 0.26951, and the list L_2 that has an area occupation of at least 0.40643. For the remaining lists, the area occupation is at least 0.53901. This is stated in the next proposition. As one can see, in general, the area occupation becomes better for small circles.

Proposition 2 *In Table 1 it is indicated the least area occupation when one packs k circles of radius in the interval $(\rho_{k+1}, \rho_k]$ in a bin, when $k \leq 8$. For $k \geq 4$, the area occupied by k circles in the list L_k is at least* 0.53901.

To see an example of an algorithm that takes advantage of optimum configurations for equal sized circles, consider an algorithm that subdivides the input list L in two sublists, L' and L''. These lists are defined as follows:

$$\begin{aligned} L' &= \{i \in L : \rho_3 < r_i \leq \rho_1\} \text{ and} \\ L'' &= L \setminus L'. \end{aligned}$$

According to Proposition 2, we can obtain a packing $\mathcal{P}'$ of L' with an area occupation of at least 0.26951 and a packing $\mathcal{P}''$ of L'' with an area occupation of at least 0.53901, except perhaps in the last bins of each packing. The following inequalities can be proved for these packings:

Table 1 Values of ρ_k and the least area occupied by k circles that belong to L_k

L_k	$\rho_{k+1} <$ radius $\leq \rho_k$	area of k circles
L_1	$\rho_2 = \frac{1}{2+\sqrt{2}} \approx 0.292893$	0.26951
L_2	$\rho_3 = \frac{2}{4+\sqrt{2}+\sqrt{6}} \approx 0.254333$	0.40643
L_3	$\rho_4 = \frac{1}{4} \approx 0.250000$	0.58905
L_4	$\rho_5 = \frac{\sqrt{2}-1}{2} \approx 0.207107$	0.53901
L_5	$\rho_6 = \frac{6\sqrt{13}-13}{46} \approx 0.187681$	0.55330
L_6	$\rho_7 = \frac{4-\sqrt{3}}{13} \approx 0.174458$	0.57369
L_7	$\rho_8 = \frac{1}{2+\sqrt{2}+\sqrt{6}} \approx 0.170541$	0.63959
L_8	$\rho_9 = \frac{1}{6} \approx 0.166667$	0.69813

$$
\begin{aligned}
|\mathcal{P}'| &\leq \frac{\text{Area }(L')}{0.26951} + \beta' \text{ and} \\
|\mathcal{P}''| &\leq \frac{\text{Area }(L'')}{0.53901} + \beta'',
\end{aligned}
$$

where β' and β'' are constants, given by the number of bins without the given area occupation for the corresponding packings.

For the sublist L', we can also obtain an optimum packing in polynomial time. To see this, note that we cannot pack three circles of L' in a same bin, as each of these circles has radius greater than ρ_3. So, any bin in an optimum packing of L' has at most two circles. Thus, we can obtain an optimum packing by finding one that maximizes the number of bins with two circles. For that, we construct a graph whose vertices correspond to the circles, and two vertices are adjacent if the corresponding circles can be packed in a same bin. The optimum packing can be obtained by computing a matching of maximum cardinality in this graph. Each edge of the maximum matching indicates a pair of circles that should be packed together. Therefore, for such packing, we have

$$|\mathcal{P}'| = \text{OPT}(L') \leq \text{OPT}(L).$$

Let $\mathcal{A}$ be the algorithm that generates a packing $\mathcal{P}$ obtained by concatenating packings $\mathcal{P}'$ and $\mathcal{P}''$, as we have defined. In the following proposition we show an asymptotic approximation factor of this algorithm.

Proposition 3 *Algorithm $\mathcal{A}$ mentioned above is an asymptotic* 2.3553*-approximation algorithm for the* CBP *problem.*

Proof Let $N' = |\mathcal{P}'| - \beta'$ and $N'' = |\mathcal{P}''| - \beta''$. The result follows when $N' \leq 0$ or $N'' \leq 0$. Therefore, consider the case when $N' > 0$ and $N'' > 0$. As $\text{OPT}(L) \geq |\mathcal{P}'|$ and $\text{OPT}(L) \geq$ Area (L), we have

$$
\begin{aligned}
\text{OPT}(L) &\geq \max\{|\mathcal{P}'|, \text{ Area }(L)\} \\
&\geq \max\{N', 0.2695N' + 0.539N''\}.
\end{aligned}
$$

Taking $\beta = \beta' + \beta''$, we have

$$
\begin{aligned}
|\mathcal{P}| &= N' + N'' + \beta \\
&\leq \frac{N' + N''}{\max\{N', 0.2695N' + 0.539N''\}} \text{OPT}(L) + \beta.
\end{aligned}
$$

The factor 2.3553 follows by analysing the two possible cases the maximum is attained in the denominator. This completes the proof. $\square$

The previous idea can be improved considering that we can obtain an almost optimum packing for any set of circles with radii at least a certain constant ε—one of the ingredients of the APTAS we consider in the next section—, combined with the fact that we can obtain area occupation that approaches the density of equal circles, of $\pi/\sqrt{12}$, for sufficiently small circles, using the algorithm mentioned in Sect. 3.2.

Using the same technique of the algorithm presented in Proposition 3, and the algorithm to pack small circles, it is possible to obtain an algorithm with an asymptotic approximation factor that can be made as close as we want to

$$\frac{N' + N''}{\max\{N',\ 0.2695N' + \pi/\sqrt{12}N''\}} \leq 1.8055.$$

We have discussed two approaches: the first, based on the idea of subdividing the input list; and the second, based on the idea of subdividing the bins. In both cases, the analyses were based on the least area occupation and partial optimal packings of the used bins.

In the next section, we consider an algorithm that combines ideas of sublists and ideas of subdivision of the bins, in a more sophisticated way to obtain an APTAS.

3.4 An APTAS for the CBP problem

In this section, we deal with the main ingredients used in the APTAS for the CBP problem that is presented in [48]. One of the main ingredients is a circle packing algorithm that, given a list of circles L, with constant number of circles of bounded radii, decides if L can be packed into a rectangle of size $w \times h$. This is presented in the next section, and is used as a subroutine of the PTAS presented in the subsequent section.

3.4.1 Packing a constant number of large circles within a bin

The authors first consider the problem of packing a list of circles $L = \{1, \ldots, n\}$, with constant number n of items to be packed into a bin of size $w \times h$, where each circle i has radius r_i that is at least a constant δ, and $2r_i \leq \min\{w, h\}$. The authors model this problem as a semi-algebraic quantifier elimination problem, to find real numbers $x_i, y_i \in \mathbb{R}_+$, for each $i \in L$, where (x_i, y_i) corresponds to the position where the center of the circle i is packed, conditioned to the following constraints:

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2 \quad \text{for } i,j \in L, \text{ and } i \neq j; \tag{1}$$

$$r_i \leq x_i \leq w - r_i \quad \text{for } i \in L; \text{ and} \tag{2}$$

$$r_i \leq y_i \leq h - r_i \quad \text{for } i \in L. \tag{3}$$

The first constraints guarantee that two distinct circles do not overlap, and the next two constraints guarantee that any circle is totally packed within a bin. The set of solutions that satisfy the above system is a semi-algebraic set in the field of the real numbers. As the number of variables and constraints are constants, the existential problem of the above system can be solved in polynomial time by quantifier elimination algorithms [2, 30]. Since a point in a semi-algebraic set may potentially be irrational, it is possible to obtain an approximate rational solution of arbitrary precision in polynomial time. To avoid circle intersections or that a circle stays partially outside the bin, the authors showed that it is possible to convert this approximate solution into a feasible packing within an augmented bin. To achieve this, the circles are moved by a small amount, and one of the bin sizes (e.g., its height) is increased by a small factor $\xi > 0$. The time complexity depends polynomially on $\log 1/\xi$, and the parameter ξ can be given as part of the problem instance. We do not present here the details on how to transform a solution with irrational coordinates to one with rational coordinates that corresponds to a packing into an augmented bin by a desired small factor $\xi > 0$. These results are stated in some of the forthcoming lemmas and theorems on augmented bins. We refer the reader to [48] for details on how this can be done.

3.4.2 Packing a set of large items with constant number of different sizes

Consider now that the input list L has only "large" circles (that is, $r_i \geq \delta > 0$, for each $i \in L$) and the number of different radii $s_1, \ldots, s_k$ is bounded by a constant (that is, each circle in L has radius s_j, for some $j \in \{1, \ldots, k\}$).

We use the term *pattern* to refer to a packing of a set of circles $S \subseteq L$ into an (augumented) bin. For our purpose, we are interested in patterns for different set of circles; and we consider that patterns corresponding to a same set of circles are equal. As the circles in L are large, the maximum number of circles in a bin is bounded by a constant, and therefore, we can enumerate all potential sets of circles that can be packed in a bin, and consequently, obtain the patterns, in augmented bins, with the algebraic quantifier elimination algorithm for the feasible ones. Denote by $\mathcal{P}_1, \ldots, \mathcal{P}_K$ all possible patterns, where K is constant, and by a_{ij} the number of circles of radius s_i in pattern $\mathcal{P}_j$. If we define an integer variable x_j representing the number of patterns $\mathcal{P}_j$ used in the solution and by n_i the number of circles of radius s_i in L, the following ILP (integer linear program) solves the problem.

$$\min \sum_{j=1}^{K} x_j \tag{4}$$

$$\sum_{j=1}^{K} a_{ij} x_j = n_i \quad \text{for } i = 1, \ldots, k \tag{5}$$

$$x_j \in \mathbb{Z}_+ \quad \text{for } j = 1, \ldots, K. \tag{6}$$

As the ILP (4–6) has a constant number of variables, it can be solved in polynomial time [14, 40]. In the following lemma we state for what kind of instances the corresponding ILP can be solved in polynomial time, and find a packing into an augmented bin.

Lemma 3 *Given an instance* (L, w, h) *for the* CBP *problem, and a constant* $\xi > 0$, *where L consists of circles with radii at least a constant, and the number of different* *radii is bounded by a constant, we can obtain in polynomial time a packing of L using at most* $\mathrm{OPT}_{w\times h}(L)$ *bins of size* $w \times (1+\xi)h$.

3.4.3 Packing large items

Let us consider instances containing only large circles, i.e., circles with radius at least a certain $\delta > 0$. We recall that Area (δ) denotes the area of a circle with radius δ.

To obtain almost optimum packings, the authors in [48] use the so-called linear rounding technique, first presented by Fernandez de la Vega and Lucker [25], and used in the development of many approximation schemes for packing problems.

They combine this technique with the previous algorithm for packing instances with large circles and bounded number of different radii and obtain an approximation scheme for the packing of large items in augmented bins.

Let ε be a small positive constant. Before applying this technique, it is first verified if the number of circles is at most $K = \lceil 2/(\varepsilon \text{ Area } (\delta)) \rceil$, in which case the algorithm returns the packing obtained from the algorithm stated in Lemma 3. Otherwise, it performs the following steps.

1. Sort L in non-increasing order of the radii of its n circles, and greedily partition L into groups each one with $Q = \lfloor n\varepsilon \text{ Area } (\delta) \rfloor$ consecutive circles, except perhaps the last group, which may have fewer circles.
2. Let L' the list obtained from L rounding down the radii of the circles in each group to the smallest radius in the corresponding sublist.
3. Use the algorithm from Sect. 3.4.2 to obtain an optimum packing $\mathcal{P}'$ in augmented bins for the list L'.
4. Except for the circles in the first group, denoted by L_0, it is possible to map each circle in $L \setminus L_0$ to a different circle in $\mathcal{P}'$, with non-smaller radius. For each circle in $\mathcal{P}'$, replace it with the mapped circle in $L \setminus L_0$, removing non-mapped circles. Denote by $\mathcal{P}''$ the obtained packing of $L \setminus L_0$.
5. Let $\mathcal{P}_0$ be a packing of L_0, obtained by packing each circle in L_0 in one bin.
6. Return the packing $\mathcal{P}_0 \| \mathcal{P}''$.

Note that the size of packing $\mathcal{P}''$ is at most $\mathrm{OPT}(L)$, as it was obtained from an optimum packing for the items of L with radii rounded down. For the list L_0, note that it has at most εn Area (δ) bins and using the area lower bound, we have $|\mathcal{P}_0| \le \varepsilon n \text{ Area } (\delta) \le \varepsilon \text{ Area } (L) \le \varepsilon\, \mathrm{OPT}(L)$. Therefore, the final packing, which is the concatenation of packings $\mathcal{P}_0$ and $\mathcal{P}''$, uses at most $(1+\varepsilon)\,\mathrm{OPT}(L)$ augmented bins.

The following lemma can be derived from the previous discussion.

Lemma 4 *Given an instance* (L, w, h) *for the* CBP *problem, each circle with radius at least* δ, *and a constant* $\xi > 0$, *there is a polynomial-time algorithm that packs L into at most* $(1+\varepsilon)\,\mathrm{OPT}_{w\times h}(L)$ *bins of size* $w \times (1+\xi)h$.

3.4.4 An APTAS for the CBP problem

The algorithm presented in the previous subsection obtains almost optimum packings for most of the practical situations, as the constant δ, that bounds the radii of large circles can be chosen arbitrarily small. On the other hand, it is difficult to pack the small circles in the space not occupied by the large circles and also maintain the packing close to optimum. In fact, as the separation of large and small circles are based only on a threshold δ, the size of large circles may be very close to the small ones. In this case, the packing of the large circles may not leave sufficient unused space to accommodate the small ones.

The key idea of the mentioned APTAS is to obtain a partition of the input list L into a gap-structured partition, defined as follows.

Definition 1 Let $I = (L, w, h)$ be an instance for the CBP problem. We say that L has a *gap-structured partition* if it can be partitioned into sets $H_t, S_0, \dots, S_m$, and there exists bin sizes $B_0, \dots, B_m$ where B_0 has size $w \times h$, such that

1. $\mathrm{OPT}(H_t) = O(\varepsilon)\,\mathrm{OPT}(L)$.
2. Items in S_j are small compared to bins of size B_j, for $j \geq 1$.
3. Bins of size B_j are small compared to circles in S_{j-1}, for $j \geq 1$.
4. Any circle in S_j has radius at least a constant factor of the size of B_j.

Lemma 5 *For any instance $I = (L, w, h)$ for the circle bin packing problem, it holds that L has a gap-structured partition $(H_t, S_0, \dots, S_m)$ as defined above.*

A gap-structured partition of a list L can be obtained as follows:

1. Partition the input list L into groups $G_0, G_1, \dots$ with (decreasing) radii bounded by powers of ε^2, that is, $G_i = \{j \in L : \varepsilon^{2i} w \geq 2r_j \geq \varepsilon^{2(i+1)} w\}$ for $i \geq 0$.
2. Let $r = 1/\varepsilon$ and $H_k = \{\ell \in G_i : i \equiv k \pmod r\}$, for $0 \leq k < r$.
3. Let t be an integer such that Area $(H_t) \leq \varepsilon$ Area (L) and $0 \leq t < r$.
4. Let $S_j = \bigcup_{i=t+(j-1)r+1}^{t+jr-1} G_i$, for $j \geq 0$.
5. Define bins B_j with size $w_j \times h_j$, where $w_0 = w$, $h_0 = h$, and $w_j = h_j = \varepsilon^{2(t+(j-1)r)+1} w$, for every $j \geq 1$.

To see the construction of these sets, the input list L is partitioned into groups $G_0, G_1, \dots$, that can be organized in the following way:

$$
\begin{array}{|lccccc|c|}
 & & & & & & H_t: \\
S_0: & & & & G_0, & \dots\ G_{t-1} & G_t \\
S_1: & G_{t+1}, & G_{t+2}, & \dots & G_r & \dots\ G_{t+r-1} & G_{t+r} \\
S_2: & G_{t+r+1}, & G_{t+r+2}, & \dots & G_{2r} & \dots\ G_{t+2r-1} & G_{t+2r} \\
S_j: & G_{t+(j-1)r+1}, & G_{t+jr+2}, & \dots & G_{3r} & \dots\ G_{t+jr-1} & G_{t+jr} \\
 & & & \vdots & \vdots & & \vdots
\end{array}
\qquad (7)
$$

The circles in the groups G_i's are grouped again according to its group index modulo r. These are the sets H_k's that contain the groups in the same column. Since there are $r = 1/\varepsilon$ columns, at least one of the sets H_t has total area at most ε Area (L). This implies that $\mathrm{OPT}(H_t) \leq \varepsilon\,\mathrm{OPT}(L)$.

After removing the set H_t, each set S_j is defined as the set of circles that belongs to groups G_i such that $t + (j-1)r < i < t + jr$. Note the gap induced by the removal of the set $G_{t+jr} \subseteq H_t$ between sets S_j and S_{j+1}. Therefore, the partition of $L \setminus H_t$ into sets $S_0, \dots, S_m$ satisfies the remaining gap-structured partition conditions.

As w_{j+1} (resp. h_{j+1}) divides w_j (resp. h_j), for $j \geq 1$, any bin B_j can be perfectly partitioned into bins of size B_{j+1}. The APTAS works as follows.

1. Let $\mathcal{P}_j$ be the packing of S_j using the algorithm of Sect. 3.4.3 and using bins B_j of size $w_j \times h_j$, for $j \geq 0$.
2. For every $j \geq 0$, do:
   (a) Consider a square grid over each bin of the packing $\mathcal{P}_j$, where each empty square cell is considered a bin of size B_{j+1}.
   (b) Pack the bins of $\mathcal{P}_{j+1}$, into these empty square cells.
   (c) If there is insufficient empty cells to pack the bins of $\mathcal{P}_{j+1}$, iteratively subdivide an empty cell/bin of size $B_{j'}$, for some $j' < j$, into smaller grids until obtain a cell/bin of size B_j. If necessary, start this process with a new empty unit bin.
3. Return the packing obtained in the previous step, concatenated with a packing of H_t of size at most $\varepsilon\,\mathrm{OPT}(L)$, *e.g.* produced by Algorithm $\mathcal{A}_{\text{SP}}$ (described in Sect. 3.1).

In what follows, we discuss how the gap-structured partition $(H_t, S_0, \dots, S_m)$ can be used to obtain an APTAS. The packing of the set H_t, built in an independent way of the other sets, has small impact in the final solution. As its total area is small, we can use Algorithm $\mathcal{A}_{\text{SP}}$ to obtain a packing that uses $O(\varepsilon)\,\mathrm{OPT}(L)$ bins.

Let us now consider the sets $S_0, S_1, \dots, S_m$. The absence of circles of H_t in these sets leads to a gap between circles of S_j and S_{j+1}, such that circles of S_{j+1} are small compared to bins B_{j+1} that are also small compared to circles in S_j. Figure 3 illustrate part of a packing with a circle of S_j and the grid Gr_{j+1} that defines the bins of size B_{j+1} and circles of S_{j+1} packed in them.

Following the ideas behind the APTAS, note that bins of size B_{j+1} are much smaller than the circles in S_j. And also note that there may exist some cells at level $j+1$, that are partially used. This is the case of the cells that are partially used by circles of S_j and the algorithm may not use that empty space (these are the hatched bins in Fig. 3). In fact, the total area of these partially used cells of size $w_{j+1} \times h_{j+1}$ around a circle $c \in S_j$ are small, as they are basically composed by the cells that match the border of c, and can be bounded within a factor of ε of the area of the circle c.

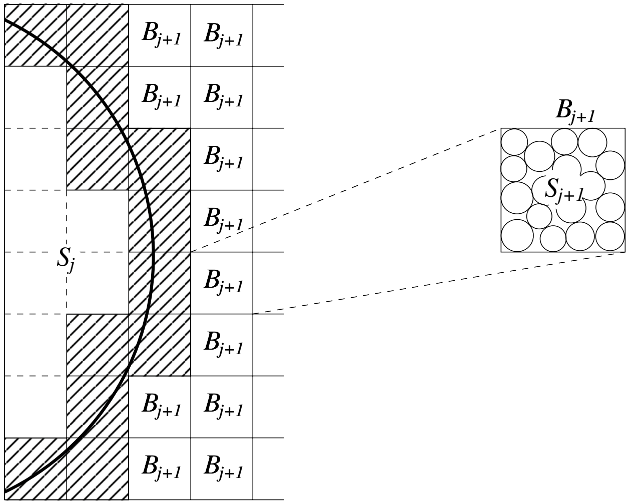


Fig. 3 Example of a packing of a circle of S_j, grid Gr_{j+1}, bins B_{j+1} and circles of S_{j+1}

Also note that although the description of the algorithm considers bins that are not augmented, the use of augmented packings does not increase the bins by too much, as we are always considering augmented bins by a factor of ξ larger than the original bins. Therefore, if the original bin is augmented by a factor of ξ, this is also sufficient to lead to augmented (sub-)bins obtained from small grid cells.

To show that the packing obtained is indeed almost optimum, we need to show that there exists an almost optimum packing that follows the same algorithmic pattern. To do so, we note that there are three important aspects that we have to consider to obtain such an optimum packing. To this end, consider an optimum packing $\mathcal{P}^*$, and suppose the following happens with respect to this packing.

A1. The items of H_t may have been packed mixed with the other circles.
A2. Bins B_{j+1} obtained from a grid over a packing of S_j may have been occupied partially by a circle of S_j.
A3. Circles of S_{j+1} may have been packed over the grid lines that define the bins B_{j+1}.

If A1 happens, remove the circles of H_t from $\mathcal{P}^*$ and concatenate the resulting packing with the packing returned by $\mathcal{A}_{\text{SP}}(H_t)$. As in the algorithm, this will increase the resulting packing by a factor of $O(\varepsilon)$ besides a constant number of bins. If A2 happens, the circles in a bin of size B_{j+1} that partially intersect a circle S_j can be moved into a new bin of the same size that does not intersect the circles in S_j. These bins intersect the circle in S_j in its frontier, and as bins B_{j+1} are very small compared to a circle of S_j, the total area of these moved circles is at most $O(\varepsilon)$ factor of the area of the circles in S_j. The total area of the new bins created in this process is at most $O(\varepsilon)$ Area (L).

Let us now consider A3. The circles of a set S_j that intersect their corresponding grid lines occupy a small amount of space compared to the area of the bin, once such circles are much smaller than the corresponding bins. In fact, the total area occupied by these circles in a bin is at most a factor of ε of the corresponding bin. On the other hand, we cannot allocate new bins by considering each set S_j independently, as we may end up allocating too much area. To deal with this situation, the algorithm first reserves a certain amount of new space corresponding to bins of size B_1 (a factor of $O(\varepsilon)$ of the Area (L)). Instead of moving circles to the new space, the algorithm uses a more sophisticated approach, moving set of circles to a new space, but also considering the space freed by large circles as new free space for smaller circles.

The APTAS result, proved in [48], can be summarized in the following theorem.

Theorem 1 *Let* (L, w, h) *be an instance for the* CBP *problem, and let* $\varepsilon > 0$ *and* $\xi > 0$ *be constants. There exists an asymptotic polynomial-time approximation scheme* $\mathcal{A}_\varepsilon^{\text{CBP}}$ *for the circle bin packing problem that finds a packing of L into at most* $(1 + \varepsilon)\,\text{OPT}_{w\times h}(L) + O(1)$ *bins of size* $w \times (1 + \xi)h$.

4 Circle strip packing

In this section, we follow the same approach used for the circle bin packing problem. We start with a more straighforward area-based algorithm and then present an approach that uses equal circles. At last, we discuss an APTAS based on the APTAS for the circle bin packing problem.

4.1 Area-based algorithms

As in the previous section, we start with a simple and natural algorithm that rounds the given circles by circumscribed squares. Coffman et al. [11] showed that if a list L consists of squares with side lengths at most $1/m$, for an integer $m \geq 1$, then NFDH algorithm (mentioned in the previous section) for the square strip packing problem has the following performance: $\text{NFDH}(L) \leq (1 + 1/m)\ \text{Area}\,(L) + 1/m$.

Based on this result, a first algorithm for the circle strip packing problem is to encapsulate each circle within a square and execute the NFDH algorithm. Since a circle inscribed in a unit square has area $\pi/4$, the asymptotic approximation ratio of NFDH° is at most $(4/\pi)(1 + 1/m)$. Thus, for $m = 1$ (the general case), this algorithm has an asymptotic factor of 2.5465.

For small circles, it is not hard to obtain a similar result with asymptotic factor that tends to $1/\tau \approx 1.1027$ as the circles become smaller, where $\tau = \pi/\sqrt{12}$ is the factor of the highest-density arrangement of equally sized circles. One can use the algorithm we discussed in the end of Sect. 3.3 to pack circles into unit bins and concatenate them to obtain a packing into a rectangular strip. This algorithm guarantees an area occupation that tends to $\pi/\sqrt{12}$, as the circles become smaller. This same guarantee holds for the occupied area in the strip.

4.2 Subdividing and rounding

In this subsection, we use a strategy to subdivide the input list into sublists, each one with similar circles. The radii of the circles in a same sublist are rounded to be all equal, and an algorithm to pack equal circles is used to obtain a packing for each sublist. The packing for the original instance is obtained by concatenating the packings obtained for each sublist.

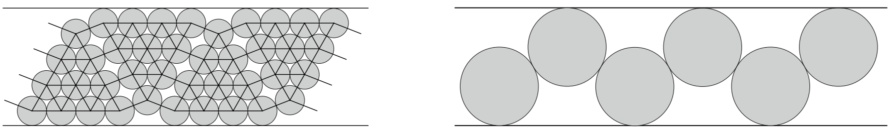
To this aim, we first consider the problem of packing equal circles. The packing of congruent circles into an infinite strip has been investigated for many years. In the seventies, Fejes-Tóth [21] raised a question about the densest packing of equal circles into a strip, and according to Brass et al. [6] and Fejes-Tóth [17], Molnár conjectured that the density d of a packing of unit circles in a parallel strip of width w satisfies

$$d \leq \frac{(n+1)(n+2)\pi}{2w\left(n+\sqrt{4-\left(w-2-n\sqrt{3}\right)^2}\right)}, \text{ where } n = \lfloor (w-2)/\sqrt{3} \rfloor .$$

If this bound is valid, it is tight, given by constructions of the type given in Fig. 4 (see Brass et al. [6], Subsection 1.7). The construction has repeated copies of a set T of $(n+1)(n+2)/2$ unit circles arranged in a triangular shape in a way that the convex hull of their centers forms an equilateral triangle of side $2n$ (in Fig. 4a, we have $n = 3$). This set T is repeated alternating its configuration upside down justified at the bottom and top of the strip. The conjecture is already proved for some values of w, in particular for values of $2 \leq w \leq 2 + 2\sqrt{3}$ [26, 39]. Translating this result to packing of congruent circles into a strip of unit width, the worst area density is given by $\pi/\sqrt{27} \approx 0.6046$ for the packing of circles of radius 1/3 (see Fig. 4b). For smaller circles, the configuration leads to better area guarantee. Based on this work, it is possible to obtain an algorithm with asymptotic approximation factor at most $\sqrt{27}/\pi \approx 1.654$. More precisely, the following holds.

Lemma 6 *Given a list L consisting of equal sized circles for the* CSP *problem, there is a polynomial-time algorithm E_δ such that $E_\delta(L) \leq (\sqrt{27}/\pi + \varepsilon)$* Area $(L) + O(\delta)$, *for any $\varepsilon > 0$.*

Now let us consider the case in which the circles have different sizes. The following algorithm, which we denote by A_ε, subdivide the input list into sublists with



(a) (b)

Fig. 4 Conjectured densest packing of equal circles in a strip circles of similar radii, and round up to a same value the radii of the circles in a same sublist. Then, it applies the corresponding Algorithm E_δ for each sublist and returns the concatenation of the produced packings. More formally, the steps are the following.

1. Subdivide the input list L into lists $L_0, L_1, \ldots$, such that
$$L_i \leftarrow \{j \in L : 1/(1+\varepsilon)^{i+1} < 2r_j \leq 1/(1+\varepsilon)^i\}.$$
2. Generate a packing $\mathcal{P}_i$ of L_i, using algorithm $E_{1/(1+\varepsilon)^i}$ and considering that each circle has diameter $1/(1+\varepsilon)^i$, for $i \geq 0$.
3. Return the concatenation $\mathcal{P}_0 \| \mathcal{P}_1 \| \ldots$.

As each of the newly obtained radii is at most $1+\varepsilon$ of the original radii, the area of each perturbed circle is at most a factor of $1+3\varepsilon$ of the area of the original circle. Summing up the area occupation for each sublist, we obtain an area density that can be made as close to $\sqrt{27}/\pi \approx 1.654$ as desired. This result can be stated as follows.

Proposition 4 *Given a list L of circles for the* CSP *problem, there exists a polynomial-time algorithm E'_ε such that $E'_\varepsilon(L) \leq (1.654 + O(\varepsilon))\,\mathrm{OPT}(L) + O(1/\varepsilon)$, for any $\varepsilon > 0$.*

It is possible to apply the same idea used in Sect. 3.3, where the input list is partitioned into large and small circles, obtaining a (quasi-)optimum packing for the large circles and better area guarantee for the remaining circles. This approach leads to an algorithm with smaller approximation factor, but its straightforward application does not lead to approximation schemes. In the next section, we consider the APTAS presented by Miyazawa et al. [48], that is also based on the gap-structured partition.

4.3 An APTAS for the CSP problem

In this section we present the main ideas of the APTAS presented by Miyazawa et al. [48], and based on the APTAS for the circle bin packing problem. First, we observe that the APTAS presented for the bin packing problem can be extended to pack circles into bins of size $w \times h$, as long as w and h are constant values.

The APTAS for the circle strip packing problem (on the input list L), which we denote by $\mathcal{A}_\varepsilon^{\mathrm{CSP}}$, proceed as follows: executes the APTAS $\mathcal{A}_\varepsilon^{\mathrm{CBP}}$ for the circle bin packing on the input list L, using bins of size $1 \times 1/\varepsilon$, and then returns a packing that concatenates the packed bins one on top of the other. The following result holds for this algorithm.

Theorem 2 *Given a list L of circles for the* CSP *problem, there exists a polynomial-time algorithm $\mathcal{A}_\varepsilon^{\mathrm{CSP}}$ that packs L into a rectangular strip of unit width and infinite height such that $\mathcal{A}_\varepsilon^{\mathrm{CSP}}(L) \leq (1+\varepsilon)\,\mathrm{OPT}_{\mathrm{CSP}}(L) + O(1/\varepsilon)$.*

The proof of the above inequality follows from the close relation between packings for the circle strip packing and for the bin packing version. Denote the minimum number of bins of size $1 \times 1/\varepsilon$ to pack L as $\mathrm{OPT}^B(L)$ and the minimum height of a packing of L into a rectangular strip of width 1 as $\mathrm{OPT}^S(L)$.

Given an optimum packing $\mathcal{P}_S^*$ for the strip packing problem, we can obtain a packing for the bin packing version with bins of size $1 \times 1/\varepsilon$, in the following way: cut the packing $\mathcal{P}_S^*$ with horizontal lines at height $\ell_0, \ell_1, \dots$ where ℓ_i cuts the packing $\mathcal{P}_S^*$ at height i/ε, for $i \geq 0$. Then, generate packings into bins of size $1 \times 1/\varepsilon$ moving the packing of the circles totally contained between the lines ℓ_{i-1} and ℓ_i to a bin B_i, for $i \geq 1$. Now consider the circles crossed by the lines ℓ_i's. For each line, the crossed circles ocupy a region of width 1 and height 2, as the diameter of each circle is at most one. Therefore, we can group the circles crossed by at least $1/(2\varepsilon)$ lines into a new bin of size $1 \times 1/\varepsilon$. These last steps cause the use of at most $\lceil 2\varepsilon^2 \mathrm{OPT}^S(L) \rceil$ new bins. So,

$$\mathrm{OPT}^B(L) \leq \frac{\mathrm{OPT}^S(L)}{1/\varepsilon} + \lceil 2\varepsilon^2 \mathrm{OPT}^S(L) \rceil \leq \varepsilon(1 + 2\varepsilon)\,\mathrm{OPT}^S(L) + 1.$$

As the size $\mathcal{A}_\varepsilon^{\mathrm{CSP}}(L)$ is basically the concatenation of the bins of size $1 \times 1/\varepsilon$, we have

$$\begin{aligned}
\mathcal{A}_\varepsilon^{\mathrm{CSP}}(L) =& \frac{1}{\varepsilon}\mathcal{A}_\varepsilon^{B}(L)\\
\leq& \frac{1}{\varepsilon}((1+\varepsilon)\,\mathrm{OPT}^B_{1\times 1/\varepsilon}(L) + 2)\\
\leq& \frac{1}{\varepsilon}((1+\varepsilon)(\varepsilon(1+2\varepsilon)\,\mathrm{OPT}^S(L) + 1) + 2)\\
\leq& (1+4\varepsilon)\,\mathrm{OPT}^S(L) + \frac{4}{\varepsilon}.
\end{aligned}$$

5 Circle knapsack

In this section, we present two algorithms for the CK problem. The first, is an algorithm with approximation factor $1/3 - \varepsilon$, for any $\varepsilon > 0$, and the second algorithm is a PTAS for the version where the value of each circle is its own area.

5.1 A constant factor algorithm for the CK problem

Let us denote by $\mathcal{C}_\varepsilon^{\mathrm{CK}}$ the $(1/3 - \varepsilon)$-approximation factor algorithm for the CK problem that is based on the ideas presented by Diedrich et al. [13] for the three-dimensional box knapsack problem. We will describe it formally in what follows, but first we sketch the idea behind it.

It first selects a set of circles based only on the area and value of the circles. To this end, it uses an algorithm for the 1D Knapsack (1K) problem. Then, the set of selected circles are packed into at most three unit bins, and the packing in one of the bins with largest value is returned.

For completeness, we recall that in the 1K problem, the input is a triple (L, s, v), where L is a set of one-dimensional items, and s and v are functions (size and value, respectively): $s : L \to \mathbb{Q}^+$ and $v : L \to \mathbb{Q}^+$. The objective is to find a subset $L' \subseteq L$ such that $s(L') \leq 1$ and $v(L')$ is maximum.

Ibarra and Kim [35] designed a Fully PTAS for the 1K problem, which we denote by $\mathcal{A}_\varepsilon^{IK}$, which guarantees that $\mathcal{A}_\varepsilon^{IK}(L) \geq (1-\varepsilon)\,\mathrm{OPT}_{1K}(L)$ for any input (L, s, v).

We are now ready to describe Algorithm $\mathcal{C}_\varepsilon^{\mathrm{CK}}$ for the CK problem.

1. Let (L, v) be the input to the CK problem. Execute $\mathcal{A}_\varepsilon^{IK}$ on the input (L, s, v), where s is a function that associates with each circle in L its area. Let $L' \subseteq L$ be the set of circles returned by $\mathcal{A}_\varepsilon^{IK}(L)$. Use an augmented bin to allow approximate calculations for the area of each circle.
2. Let S_1, S_2 and S_3 be empty sets.
3. Sort the circles in the list L' in non-increasing order of their radii. For each $x \in L'$, insert x in the first set among S_1, S_2, S_3 that has the smallest total area.
4. Let S be the first set among S_1, S_2, S_3 that has the maximum total value.
5. If S has only one circle, return a packing containing it. Otherwise, return $\mathcal{A}_{\mathrm{SP}}(S)$.

Theorem 3 *Algorithm $\mathcal{C}_\varepsilon^{\mathrm{CK}}$ returns a solution for the* CK *problem into augmented bins such that*

$$\mathcal{C}_\varepsilon^{\mathrm{CK}}(L) \geq \frac{1}{3}(1 - O(\varepsilon))\,\mathrm{OPT}_{\mathrm{CK}}(L), \text{ for any input } (L, v).$$

This analysis is tight.

Proof Let (L, v) be the input to the CK problem, and let (L, s, v) be the input to $\mathcal{A}_\varepsilon^{IK}$, according to step 1 (of the description of Algorithm $\mathcal{C}_\varepsilon^{\mathrm{CK}}$). Let L' be the set of circles obtained by $\mathcal{A}_\varepsilon^{IK}(L)$. Clearly, we have $v(L') \geq (1-\varepsilon)\,\mathrm{OPT}_{\mathrm{CK}}(L)$. Let $\{S_1, S_2, S_3\}$ be the partition of L' produced in step 3. First, we need to prove that any set S_i chosen in step 4, has only one circle or has total area at most $\gamma = 0.5390\ldots$ (see Sect. 3.1). Without loss of generality, suppose that $\text{Area}\,(S_1) \geq \text{Area}\,(S_2) \geq \text{Area}\,(S_3)$. Let j be the last circle packed in S_1 and $y = \text{Area}\,(j)$. At the moment j was packed, the other two sets have total volume at least $x = \text{Area}\,(S_1 - j)$. As the total area of the circles is at most $1 + \xi$, for a constant ξ, due to the augmented bin, we have $3x + y \leq 1 + \xi$. Since $y \leq x$, the maximum value we have for $\text{Area}\,(S_1)$, given by $x + y$, is when $y = x$. Therefore, $\text{Area}\,(S_1) \leq 1/2 + \xi$; and thus, S_1 can be packed by Algorithm $\mathcal{A}_{\mathrm{SP}}$, using sufficiently small value of ξ. Since the set S chosen in step 4 has the largest value, we have $v(S) \geq (1/3)v(L') \geq (1/3)(1 - O(\varepsilon))\,\mathrm{OPT}_{\mathrm{CK}}(L)$.

To show that the ratio is tight, consider an instance consisting of 4 circles: one circle with radius 1/2 and value 1 and three equal circles with radius 0.3 and value $1/3 + \varepsilon$. In step 1, it is possible to make the approximation scheme select the three equal circles, obtaining value $1 + 3\varepsilon$. Then, these circles are packed in separate bins, and the algorithm returns a solution (with one of the 3 equal circles) of value

$1/3 + \varepsilon$. It is immediate that the optimum solution is given by the largest circle that has value 1. □

5.2 A PTAS for the area maximization version

When the values of the circles are given by their area, Lintzmayer et al. [41] presented an approximation scheme for the circle knapsack problem. The algorithm also uses (indirectly) the gap-structured partition approach used in the APTAS for the CBP problem, presented in Sect. 3.4.4. Since this algorithm uses a similar approach, we only mention the main ideas involved in this PTAS. The presentation is divided in two parts. One part for the version of multiple bins and large circles, and the second part for the main algorithm, that uses the algorithm for the first part as a subroutine.

5.2.1 Circle multiple knapsack problem with large circles

An instance of the Circle Multiple Knapsack (CMK) problem [41] consists of a tuple (L, w, h, f), where L is a list of circles and we have f bins of size $w \times h$. A solution is a pair $(L', \mathcal{P}')$, where $L' \subseteq L$ and $\mathcal{P}'$ is a packing of L' into f bins of size $w \times h$. In this case, we will also allow packing into augmented bins of size $w \times (1 + \gamma)h$, for a given constant $\gamma > 0$. In the CMK problem the objective is to find a solution $(L', \mathcal{P}')$ such that Area (L') is maximum.

Given an instance (L, w, h, f) for the CMK problem, where each circle $i \in L$ is large, that is, has radius $r_i > \delta$, Lintzmayer et al. [41] presented an algorithm that obtains a solution for large circles into augmented bins using at most $\mathrm{OPT}_{\mathrm{CMK}}(L)$ bins.

First, this algorithm runs a preprocessing of the instance so that it obtains a similar instance with constant number of different radii. To this end, it rounds down each circle radius r_i to one with radius $r'_i = \delta(1 + \varepsilon)^k$, where $\delta(1 + \varepsilon)^k \leq r_i < \delta(1 + \varepsilon)^{k+1}$, and $k \geq 0$.

Instead of packing the original circles, the algorithm packs the circles using the radii given by r', instead of r. Since the maximum number of circles in a bin is bounded by a constant, it is possible to find a value of ε sufficiently small so that we can obtain a feasible packing for the original circles into augmented bins by shifting the items by a small amount. After rounding down the radii, we end up with an instance for the CMK problem consisting only of large circles and a constant number of different sizes. The approach used to solve these type of instances follows closely the one used in Sect. 3.4.2. First, the algorithm generates all possible patterns concerning packing of circles into (augmented) bins, where two patterns are different if they pack different sets of circles. This is made by enumeration and the use of the algebraic quantifier elimination algorithm to obtain feasible packings into augmented bins. As the total number of different packing patterns is bounded by a constant, and it is possible to obtain an optimum packing in augmented bins with an integer linear program similar to the one presented in the model (4)–(6). In this case, the model has one integer variable for each pattern; a constraint to impose that each circle is not used more than its demand; and a constraint to impose that the total number of patterns is bounded by f. The objective function maximizes the total area of the chosen patterns. As this integer linear program has constant number of variables, it can be solved in polynomial time [14, 40]. The result presented by Lintzmayer et al. [41] can be summarized as follows.

Lemma 7 *Given an instance* (L, w, h, f) *for the* CMK *problem, where each circle* $i \in L$ *has radius at least a constant* $\delta > 0$, *there is a polynomial-time algorithm that finds a solution* $(L', \mathcal{P}')$ *into augmented bins such that* $\text{Area}\,(L') \geq \text{OPT}_{\text{CMK}}(L, w, h, f)$.

5.2.2 The main algorithm

As a first step, Lintzmayer et al. [41] consider that the input list has circles with total area at least some constant factor of the bin area. To this end, we can consider that the input list has total area at least $\gamma = 0.539$ (a factor of the unit bin), otherwise it is possible to pack all circles in the input list into a unit bin using the Split Packing algorithm [24], already mentioned in the Sect. 3.1. From now on, we consider that the input list has total area at least γ, which implies that the optimum is at least $\gamma/2$.

The main idea of this algorithm also relies on the technique of gap-structured partition, that we have mentioned to obtain the APTAS for the CBP problem. For the latter problem, the gap-structured partition $(H_t, S_0, \dots, S_m)$, is obtained by partitioning the input list into groups $G_0, G_1, \dots$ which are also organized into $r = 1/\varepsilon$ columns, producing sets $H_0, H_1, \dots, H_{r-1}$ where H_i contains the set of circles in the groups G_j, where $j = i \,(\text{mod}\ r)$. The idea to organize in these many sets was to obtain a set H_t that could be removed, so that we keep the gap-structured property for the remaining sets, and the packing of H_t has a small impact in the final packing.

Similarly, for the CK problem, the organization in these same sets will guarantee the existence of a set H_t that will also have small impact in an optimum solution. More precisely, if L^* is an optimum solution, there must exist $t \in [0, r-1]$ such that $\text{Area}\,(H_t \cap L^*) \leq \varepsilon\, \text{OPT}_{\text{CK}}(L)$, as we have $r = 1/\varepsilon$ sets H_i's. If we know such set H_t, we can simply remove these circles from the input list and the remaining circles will still lead to a quasi-optimum solution.

Unlike the CBP problem, we cannot identify the set H_t in advance, as the optimum solution for an instance of the CK problem may not include all circles of the input list. On the other hand, we can test each possible t, obtaining a solution $\mathcal{T}_t$ produced from the set $L \setminus H_t$. At last, the algorithm returns the best obtained solution among the sets $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{r-1}$. Proceeding this way, the algorithm will produce a quasi-optimal solution equal to or as good as the solution obtained excluding the correct set H_t.

From now on, we assume that the correct set H_t was removed from the input list and we are left with the sets $S_0, S_1, \dots, S_m$ with the property of gap-structured partition. The algorithm also considers bins of sizes $B_0, \dots, B_m$, for which circles in S_j are small compared to bins of size B_j, which are small compared to circles in S_{j-1}, for $j \geq 1$, as made for the APTAS for the CBP problem.

The algorithm starts with a set F_0 containing one bin B of type B_0, and setting its availability as $a_0 = 1$. The algorithm iterates for each $j \geq 0$, packing circles of S_j into a_j bins of size B_j maximizing the area allocation in each iteration. For each iteration $j \geq 0$, the algorithm finds an optimum packing of S_j into the a_j (augmented) bins of F_j, using the algorithm of Sect. 5.2.1. Then, it updates the number $a_{j+1} \leq n$ of available bins for the next iteration. To this end, the algorithm generates a grid Gr_{j+1} with cells of size B_{j+1} over the used bins of F_j and let F_{j+1} be the grid cells that do not intersect circles of S_j, added with the number of bins of size B_{j+1} that can be made available from the previous iterations. This process stops when all non-empty sets S_j's have been processed.

An important aspect to use gap-structured instances is the fact that after packing circles from the sets $S_0, \ldots, S_j$, the remaining circles of $S_{j+1}, \ldots, S_m$ and bins $B_{j+1}, \ldots, B_m$ are like sand that can be more freely alocated around the spaces between the larger circles. Moreover, after packing circles in the set S_j, the bins from the grid Gr_{j+1} that are partially covered by circles from S_j may be disregarded, as they will represent a small fraction of the packed circles of the set S_j.

To conclude the analysis, it is also needed to show the existence of an optimum or quasi-optimum packing respecting this type of structured packing separated into bins. We do not need to show the existence of an optimum packing, but a quasi-optimal packing whose value differs from the optimum by a factor of at most ε. We will not go into details, as the process is similar to the one considered to show that an optimum packing for the CBP problem packs the selected circles of S_j into bins of size B_j, for $j \geq 0$, using a small amount of additional space. For the CK problem, this additional space can be discarded, as it represents a small amount of space. The PTAS result can be sumarized in the following theorem.

Theorem 4 *Given an instance (L, w, h) for the* CK *problem, and a constant $\epsilon > 0$, there exists a polynomial-time approximation scheme $\mathcal{A}_{\varepsilon}^{\mathrm{CMK}}$ that obtains a solution $(L', \mathcal{P}')$ in an augmented bin such that $v(L') \geq (1-\varepsilon)\,\mathrm{OPT}_{\mathrm{CK}}(L, w, h)$.*

6 Online algorithms

In this section, we present online algorithms for both the circle strip packing and the circle bin packing problems, to give a flavour of them. These algorithms are based on the algorithms presented in Sects. 4.1 and 4.2 .

6.1 Online circle strip packing

6.1.1 Inscribing each circle in a square

We first consider straightforward algorithms using existing online algorithms to pack squares. Baker and Schwarz [1] presented an online version of the Next Fit algorithm for the rectangular online strip packing problem, they called Next Fit Shelf with parameter $0 < p < 1$, denoted by NFS_p. This algorithm packs, if possible, a rectangle of height h, where $p^{k+1} < h \le p^k$, into the last level of height p^k. If this is not possible, a new empty level of height p^k is created on top of the current packing, before packing the next rectangle. For the packing of squares, the next result is valid.

Lemma 8 *Given a list L of squares, we have that* $\mathrm{NFS}_p(L) \le \frac{2}{p}$ Area $(L) + \frac{1}{p(1-p)}$.

Given an instance for the circle strip packing problem, we can inscribe each circle in a square and use the algorithm NFS_p. Let us call NFS_p° the resulting algorithm. Since the area of a circle inscribed into a square is $\pi/4$ of the square, the NFS_p° algorithm can have an asymptotic factor that can be made as close to $8/\pi \approx 2.5465$ when $p \to 1$.

6.1.2 Partitioning the list of circles into sublists of circles of fixed size

Now we show that it is possible to adapt the algorithm presented in Sect. 4.2, for the (offline) strip packing problem, to the online case. Recall that this algorithm subdivides the input list of circles according to their diameters: it rounds up the diameter in the interval $(1/(1+\varepsilon)^{i+1}, 1/(1+\varepsilon)^i]$ to $1/(1+\varepsilon)^i$, for $i \ge 0$. Then, the input list is partitioned into sublists, each one with circles of the same size and the final packing is a concatenation of packings of equal circles. Although the above description does not seem suited for an online algorithm, we may adapt this algorithm to the online case.

The idea is to pre-reserve sufficiently large regions to pack circles of the same size in an online fashion within each region and booking new regions whenever a region cannot receive more items. More precisely, the adapted algorithm packs equal circles of diameter $1/(1+\varepsilon)^i$ into active regions, which are bins of width 1 and height $1/(\varepsilon(1+\varepsilon)^i)$. When a region cannot receive another circle, it is closed and a new active region is opened for the circles of the same size. When a region becomes closed, its area occupation is close to the best possible for circles of the same size. The active regions may have small area occupation, but a simple calculation shows that it is possible to bound the total area of all active regions to $O(1/\varepsilon^2)$. Moreover, the unused fraction in each closed bin, due to incomplete packings at the top and bottom of each bin, is at most $O(\varepsilon)$.

Considering the worst area guarantee for the packing of regions of large and small circles, it is possible to guarantee an area occupation close to $\pi/\sqrt{27} \approx 0.6046$. Thus, we obtain an algorithm with asymptotic performance bound $\sqrt{27}/\pi \approx 1.654$.

Lemma 9 *Given a list of circles L, there is an online algorithm* $\mathcal{A}_\varepsilon$ *for the* CSP *problem, where* $\varepsilon > 0$, *such that* $\mathcal{A}_\varepsilon(L) \le (1.6538 + \varepsilon)$ Area $(L) + O(1/\varepsilon^2)$.

6.2 Online circle bin packing

An important technique used to prove bounds for bin packing problems is the use of a weighting function introduced by Ullman [53] and used to prove approximation factors for many packing algorithms [9, 10, 49]. Consider any instance (L, B) of a bin packing problem variant, where a list of items L must be packed into the minimum number of recipients of type B, and an algorithm $\mathcal{A}$ for this same problem variant. The idea resumes to obtain a weighting function $W_{\mathcal{A}} : L \to \mathbb{R}$ such that

(a) $\mathcal{A}(L) \leq W_{\mathcal{A}}(L) + C$,
(b) $W_{\mathcal{A}}(L)/\mathrm{OPT}(L) \leq \alpha$,

where $C \geq 0$ and $\alpha \geq 1$ are constants. When these two conditions are valid, we conclude that $\mathcal{A}$ has asymptotic approximation factor α, as we have

$$\mathcal{A}(L) \leq W_{\mathcal{A}}(L) + C \leq \alpha\,\mathrm{OPT}(L) + C.$$

Note that to prove an asymptotic approximation factor α for an algorithm $\mathcal{A}$, it is sufficient to obtain a weighting function W such that (i) any set of items that is packable in a bin has total weight at most α; (ii) each bin generated by algorithm $\mathcal{A}$, except for a constant number of them, has weight at least 1. In this case, (a) is valid because the weight of the packing produced by $\mathcal{A}$ is at least 1, except perhaps in a constant number of bins, and given an optimum packing $\mathcal{O}^*$, (b) is valid because $W(L) = \sum_{B \in \mathcal{O}^*} W(B) \leq \alpha\,\mathrm{OPT}(L)$.

To see a first example of analysis using a weighting function, consider the *offline* $\mathcal{A}_{\text{SP}}^{\text{CBP}}$ algorithm, presented in Sect. 3.1. This algorithm uses Algorithm $\mathcal{A}_{\text{SP}}$ to guarantee that each bin has an area occupation of at least $\gamma/2$, except perhaps the last bin. Recall that $\gamma = 0.5390\ldots$ is the upper bound area for any list S for which $\mathcal{A}_{\text{SP}}$ is guaranteed to pack S in only one bin [24]. Let $W_{\mathcal{A}_{\text{SP}}^{\text{CBP}}}$ be a weighting function defined as $W_{\mathcal{A}_{\text{SP}}^{\text{CBP}}}(c) = \frac{2}{\gamma}\,\mathrm{Area}\,(c)$, for any circle c. As each bin produced by the Algorithm $\mathcal{A}_{\text{SP}}^{\text{CBP}}$ has an area occupation of at least $\gamma/2$, except possibly for the last bin, the weighting function guarantees a total weight of at least 1 for any bin produced by Algorithm $\mathcal{A}_{\text{SP}}^{\text{CBP}}$, except possibly for the last bin. Moreover, the maximum weight of a bin is at most $2/\gamma$, as the total area of a bin is at most 1. This shows that algorithm $\mathcal{A}_{\text{SP}}^{\text{CBP}}$ has asymptotic approximation factor $2/\gamma \approx 3.7105$. As an exercise, the reader can prove the bounds proved in Sect. 3.3 using appropriate weighting functions.

Using this technique, Hokama et al. [34] presented an algorithm that subdivides the input list L into M sublists, $L_1, \ldots, L_{M-1}, L_M$, for a certain constant integer M, where L_i has the circles in L with radius in $(\rho_{i+1}, \rho_i]$, mentioned in Sect. 3.3, for $i = 1, \ldots, M-1$. The last sublist L_M has the remaining circles. Here, the circles in the sublist L_M are called *small* while the circles in sublist L_i, for $i = 1, \ldots, M-1$ are called *big* circles of type i.

For the packing of big circles of type i, the algorithm packs i circles in each bin, except perhaps in the last bin. To this end, the algorithm maintains one active bin for each type of big circle, and whenever a type i circle cannot be packed in the corresponding active type i bin, the algorithm closes this bin and packs the circle in a newly created active type i bin. For the circles of sublist L_M, it uses the algorithm presented in Sect. 3.2, that can be made to execute in online fashion. The weighting function used for this algorithm, gives a weight of $1/i$ for each big circle of type i. This guarantees that each closed bin for these circles has weight at least 1, as the algorithm packs i circles of type i in each closed bin of type i.

For each small circle c in sublist L_M, the algorithm defines a weight that is a bit larger than Area $(c)\sqrt{12}/\pi$. As the sublist L_M has circles with small radius and the algorithm presented in Sect. 3.2 obtains an area occupation that becomes close to $\pi/\sqrt{12}$ as M becomes larger, it also leads to total weight of at least 1 for each closed bin with small circles.

To obtain an upper bound of 2.4394 for the asymptotic approximation factor of this algorithm, they used a combined integer programming and constraint programming approach to show that any set S that is packable in one bin has total weight at most 2.4394. The above algorithm can be implemented to use at most a constant number of active bins, in which case it is said to be a *bounded online algorithm*. Using similar approach, they also showed that any algorithm using a constant number of active bins must have an asymptotic approximation factor at least 2.2920.

Further, Lintzmayer et al. [42], using the same approach and improving the occupation ratio for a class of small circles, improved the upper bound to 2.3536.

These authors also presented online approximation algorithms for packing circles into isosceles right triangle bins, with asymptotic approximation factor bounded by 2.5490 and a lower bound of 2.1193. They also considered a variant where the items must be packed in an online fashion, but are allowed to be reorganized inside the bin (but cannot leave the bin where they are packed). For this variant, they presented algorithms that combine large circles of different types and use a generalized form of the weighting method [15, 49], using two weighting functions. They obtained circle bin packing algorithms with asymptotic approximation factors bounded by 2.311 for the case of square bins and by 2.51 for isosceles right triangle bins.

7 Sphere packing problems

All problems we have considered for circles have a corresponding version for the three-dimensional case, or higher dimensions. Most of the results we have shown here can be extended. In fact, the asymptotic approximation schemes presented by Miyazawa et al. [48] are also valid for the d-dimensional bin and strip packing versions, for any constant dimension d. Moreover, their work can also be generalized to other bins/items of different shapes, like ellipses, regular polygons, or even L_p-norm spheres. Lintzmayer et al. [42] also presented online algorithms for the problem of packing spheres into cubes, presenting an algorithm with asymptotic approximation factor bounded by 2.5316 and a lower bound of 2.7707 for any bounded space algorithm.

8 Concluding remarks

When considering approximation algorithms for a problem, an important question that one may consider is about the best possible factor one can achieve for this problem. We recall that for all three packing problems addressed here, we presented approximation schemes when augmented bins are allowed, for these problems and some corresponding variants. These are the best factors one can obtain considering that these problems are NP-hard. On the other hand, most of these algorithms are purely theoretical, as they have time complexity given by polynomials that are prohibitive in practice for most of the values of ε. The other algorithms, which we mentioned, and that are not approximation schemes, can be implemented to have practical running times, but of course with worse approximation factors. It would be interesting to have further algorithms with better approximation factors but still with practical running times.

It would also be interesting to consider approximation algorithms for other packing problems where items and/or bins have circular/spherical shapes. There are some interesting results that could be explored in the development of approximation algorithms. As we mentioned, Fekete et al. [23] established optimal worst-case density for packing disks into a disk, making use of a number of computer-assisted proofs. More recently, Fekete et al.[22] obtained similar results for packing squares into a disk. Becker et al. [4] give a good overview of related works, and a number of techniques.

Acknowledgements The authors are grateful to the referees for many valuable suggestions, references and comments.

Funding Research partially supported by CNPq (Proc. 314366/2018-0, 423833/2018-9, 311892/2021-3) and by FAPESP (Proc. 2015/11937-9, 2016/01860-1)

References

1. Baker, B.S., Schwarz, J.S.: Shelf algorithms for two-dimensional packing problems. SIAM J. Comput. **12**, 508–525 (1983)
2. Basu, S., Pollack, R., Roy, M.F.: On the combinatorial and algebraic complexity of quantifier elimination. J. ACM **43**(6), 1002–1045 (1996)
3. Baur, C., Fekete, S.: Approximation of geometric dispersion problems. Algorithmica **30**, 451–470 (2001)
4. Becker, A., Fekete, S., Keldenich, P., Morr, S., Scheffer, C.: Packing geometric objects with optimal worst-case density. In: Proc. of the 35th International Symposium on Computational Geometry (SoCG 2019), pp. 63:1–63:6 (2019)
5. Boll, D.W., Donovan, J., Graham, R.L., Lubachevsky, B.D.: Improving dense packings of equal disks in a square. Electron. J. Comb. **7**(R46), 9 (2000)
6. Brass, P., Moser, W., Pach, J.: Research Problems in Discrete Geometry. Springer, Berlin (2005)
7. Christensen, H.I., Khan, A., Pokutta, S., Tetali, P.: Approximation and online algorithms for multidimensional bin packing: a survey. Comput. Sci. Rev. **24**, 63–79 (2017)
8. Chung, F.R.K., Garey, M.R., Johnson, D.S.: On packing two-dimensional bins. SIAM J. Algebraic Discrete Methods **3**, 66–76 (1982)

9. Coffman, E.G., Jr., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing: an updated survey. In: Ausiello, G., Lucertini, M., Serafini, P. (eds.) Algorithms Design for Computer System Design, pp. 49–106. Springer, New York (1984)
10. Coffman, Jr., E.G., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing: a survey. In: Hochbaum, D. (ed.) Approximation Algorithms for NP-hard Problems, chap. 2, pp. 46–93. PWS (1997)
11. Coffman, E.G., Jr., Garey, M.R., Johnson, D.S., Tarjan, R.E.: Performance bounds for level oriented two-dimensional packing algorithms. SIAM J. Comput. **9**, 808–826 (1980)
12. Demaine, E.D., Fekete, S.P., Lang, R.J.: Circle packing for origami design is hard. In: Proceedings of the 5th International Conference on Origami in Science, pp. 609–626 (2010)
13. Diedrich, F., Harren, R., Jansen, K., Thöle, R., Thomas, H.: Approximation algorithms for 3d orthogonal knapsack. J. Comput. Sci. Technol. **23**(5), 749–762 (2008)
14. Eisenbrand, F.: Fast Integer Programming in Fixed Dimension. In: Di Battista, G., Zwick, U. (eds.) Algorithms - ESA 2003. Lecture Notes in Computer Science, vol. 2832, pp. 196–207. Springer, Berlin (2003)
15. Epstein, L.: Two-dimensional online bin packing with rotation. Theoret. Comput. Sci. **411**, 2899–2911 (2010)
16. Erdős, P.: Gráfok páros körüljárású részgráfjairól. Mat. Lapok (N.S.) **18**, 283–288 (1967)
17. Fejes-Tóth, G.: New results in the theory of packing and covering. In: Gruber, P., Wills, J. (eds.) Convexity and its Applications. Birkhäuser, Basel (1983)
18. Fejes-Tóth, G.: Packing and covering. In: Goodman, J., O'Rourke, J., Tóth, C.D. (eds.) Handbook of Discrete and Computational Geometry, 3rd edn. CRC Press, Boca Raton (2017)
19. Fejes-Tóth, G., Kuperberg, W.: A survey of recent results in the theory of packing and covering. In: Pach, J. (ed.) New Trends in Discrete and Computational Geometry, Algorithms and Combinatorics, vol. 10, pp. 251–279. Springer, Berlin (1993)
20. Fejes-Tóth, L.: Über einen geometrischen Satz. Math. Z. **46**, 83–85 (1940)
21. Fejes-Tóth, L.: Parasites on the stem of a plant. Am. Math. Mon. **78**(5), 528–529 (1971)
22. Fekete, S., Gurunathan, V., Juneja, K., Keldenich, P., Scheffer, C.: Packing squares into a disk with optimal worst-case density. In: Proc. of the 37th International Symposium on Computational Geometry (SoCG 2021), pp. 35:1–36:16 (2021)
23. Fekete, S., Keldenich, P., Scheffer, C.: Packing disks into disks with optimal worst-case density. In: Proc. of the 35th International Symposium on Computational Geometry (SoCG 2019), pp. 35:1–35:19 (2019)
24. Fekete, S.P., Morr, S., Scheffer, C.: Split packing: algorithms for packing circles with optimal worst-case density. Discrete Comput Geom **61**, 562–594 (2019)
25. Fernandez de la Vega, W., Lueker, G.S.: Bin packing can be solved within $1 + \epsilon$ in linear time. Combinatorica **1**(4), 349–355 (1981)
26. Füredi, Z.: The densest packing of equal circles into a parallel strip. Discrete Comput. Geom. **6**, 95–106 (1991)
27. Garey, M.R., Graham, R.L., Ullman, J.D.: Worst-case analysis of memory allocation algorithms. In: Proc. 4th Annual ACM Symp. on the Theory of Computing, pp. 143–150 (1972)
28. Graham, R., Lubachevsky, B.: Repeated patterns of dense packings of equal disks in a square. Electron. J. Comb. **3** (1996)
29. Graham, R.L.: Bounds for certain multiprocessor anomalies. Bell Syst. Tech. J. **45**, 1563–1581 (1966)
30. Grigor'ev, D.Y., Vorobjov, N.N., Jr.: Solving systems of polynomial inequalities in subexponential time. J. Symb. Comput. **5**(1–2), 37–64 (1988)
31. Hales, T.C.: A proof of the Kepler conjecture. Ann. Math. **162**(3), 1065–1185 (2005)
32. Hales, T.C., Adams, M., Bauer, G., Dang, T.D., Harrison, J., Hoang, L.T., Kalisxyk, C., Magron, V., MClaughlin, S., Nguyen, T.T., Nguyen, Q.T., Nipkow, T., Obua, S., Pleso, J., Rute, J., Solovyev, A., Ta, T.H.A., Tran, N.T., Trieu, T.D., Urban, J., Vu, K., Zumkeller, R.: Formal Proof of the Kepler Conjecture. Forum Math Pi **5**, e2 (2017)
33. Hifi, M., M'Hallah, R.: A literature review on circle and sphere packing problem: models and methodologies. Adv. Oper. Res. (2009)
34. Hokama, P., Miyazawa, F.K., Schouery, R.C.S.: A bounded space algorithm for online circle packing. Inf. Process. Lett. **116**(5), 337–342 (2016)
35. Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the knapsack and sum of subset problems. J. ACM **22**, 463–468 (1975)

36. Johnson, D.S.: Near-optimal bin packing algorithms. Ph.D. thesis, MIT, Cambridge, MA (1973)
37. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci. **9**(3), 256–278 (1974)
38. Kepler, J.: Strena Seu de Nive Sexangula (New Year's Gift of Hexagonal Snow). Godfrey Tampach, Frankfurt-am-Main, Germany (1611)
39. Kertész, G.: On a Problem of Parasites (in Hungarian). Master's thesis, Eötvös University, Budapest (1982)
40. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
41. Lintzmayer, C.N., Miyazawa, F.K., Xavier, E.C.: Two-dimensional knapsack for circles. In: M.A. Bender, M. Farach-Colton, M.A. Mosteiro (eds.) Proc. of the LATIN 2018: Theoretical Informatics, vol. LNCS 10807, pp. 741–754 (2018)
42. Lintzmayer, C.N., Miyazawa, F.K., Xavier, E.C.: Online circle and sphere packing. Theoret. Comput. Sci. **776**, 75–94 (2019)
43. Lodi, A., Martello, S., Monaci, M.: Two-dimensional packing problems: a survey. Eur. J. Oper. Res. **141**, 241–252 (2002)
44. Maranass, C.D., Floudas, C.A., Pardalos, P.M.: New results in the packing of equal circles in a square. Discrete Math. **142**(1–3), 287–293 (1995)
45. Markót, M.C.: Interval methods for verifying structural optimality of circle packing configurations in the unit square. J. Comput. Appl. Math. **199**(2), 353–357 (2007). Special Issue on Scientific Computing, Computer Arithmetic, and Validated Numerics (SCAN 2004)
46. Markót, M.C.: Improved interval methods for solving circle packing problems in the unit square. J. Global Optim. **81**(3), 773–803 (2021)
47. Meir, A., Moser, L.: On packing of squares and cubes. J. Comb. Theory Ser. A **5**, 116–127 (1968)
48. Miyazawa, F.K., Pedrosa, L.L.C., Schouery, R.C.S., Sviridenko, M., Wakabayashi, Y.: Polynomial-time approximation schemes for circle and other packing problems. Algorithmica **76**(2), 536–568 (2016)
49. Seiden, S.S.: On the online bin packing problem. J. ACM **49**(5), 640–671 (2002)
50. Specht, E.: Packomania. http://www.packomania.com (2021)
51. Szabó, P.G., Markót, M.C., Csendes, T., Specht, E., Casado, L., García, I.: New Approaches to Circle Packing in a Square. Springer, Berlin (2007)
52. Thue, A.: Über die dichteste Zusammenstellung von kongruenten Kreisen in einer Ebene. Christiania [Oslo] : J. Dybwad (1910)
53. Ullman, J.D.: The performance of a memory allocation algorithm. Tech. Rep. 100, Princeton University (1971)