Packing Disks into Disks with Optimal Worst-Case Density

Sándor P. Fekete[1], Phillip Keldenich[1], and Christian Scheffer[1]

1 **Department of Computer Science, TU Braunschweig, Germany**
{s.fekete,p.keldenich,c.scheffer}@tu-bs.de

Abstract

We provide a tight result for a fundamental problem arising from packing disks into a circular container: The *critical density* of packing disks in a disk is $1/2$. This implies that any set of (not necessarily equal) disks of total area $\delta \leq 1/2$ can always be packed into a disk of area 1; on the other hand, for any $\varepsilon > 0$ there are sets of disks of area $1/2 + \varepsilon$ that cannot be packed. The proof uses a careful manual analysis, complemented by a minor automatic part that is based on interval arithmetic. Beyond the basic mathematical importance, our result is also useful as a blackbox lemma for the analysis of recursive packing algorithms.

An longer version will appear in the 35th Symposium on Computational Geometry [3].

1 Introduction

Deciding whether a set of disks can be packed into a given container is a fundamental geometric optimization problem that has attracted considerable attention; see below for references. Disk packing also has numerous applications in engineering, science, operational research and everyday life, e.g., for the design of digital modulation schemes [19], packaging cylinders [1, 8], bundling tubes or cables [24, 22], the cutting industry [23], or the layout of control panels [1], or radio tower placement [23]. Further applications stem from chemistry [25], foresting [23], and origami design [13].

Like many other packing problems, disk packing is typically quite difficult; what is more, the combinatorial hardness is compounded by the geometric complications of dealing with irrational coordinates that arise when packing circular objects. This is reflected by the limitations of provably optimal results for the optimal value for the smallest sufficient disk container (and hence, the densest such disk packing in a disk container), a problem that was discussed by Kraviz [12] in 1967: Even when the input consists of just 13 unit disks, the optimal value for the densest disk-in-disk packing was only established in 2003 [7], while the optimal value for 14 unit disks is still unproven. The enormous challenges of establishing densest disk packings are also illustrated by a long-standing open conjecture by Erdős and Oler from 1961 [18] regarding optimal packings of n unit disks into an equilateral triangle, which has only been proven up to $n = 15$. For other examples of mathematical work on densely packing relatively small numbers of identical disks, see [9, 15, 5, 6], and [20, 14, 10] for related experimental work. Many authors have considered heuristics for circle packing problems, see [23, 11] for overviews of numerous heuristics and optimization methods. The best known solutions for packing equal disks into squares, triangles and other shapes are continuously published on Specht's website `http://packomania.com` [21].

For deciding whether a set of not necessarily equal disks can be packed into a square container, Demaine, Fekete, and Lang in 2010 [2] gave a proof of NP-hardness by using a reduction from 3-Partition, so we cannot expect that there is a deterministic polynomial-time algorithm for this problem.

The related problem of packing square objects has also been studied for a long time. Already in 1967, Moon and Moser [16] proved that it is possible to pack a set of squares

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 19–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

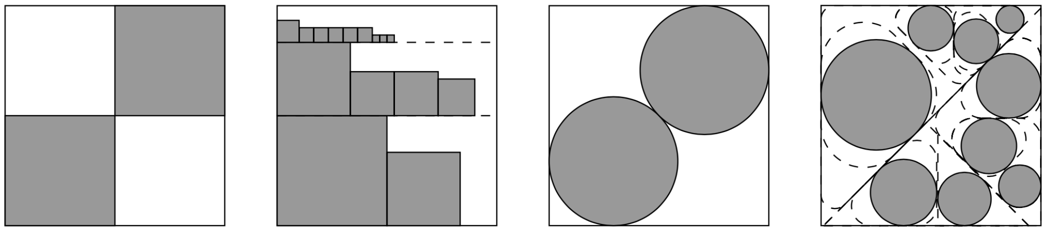

Figure 1 (1) An instance of critical density for packing squares into a square. (2) An example packing produced by Moon and Moser's shelf-packing. (3) An instance of critical density for packing disks into a square. (4) An example packing produced by Morr's Split Packing.

into the unit square in a shelf-like manner if their combined area, the sum of all squares' areas, does not exceed $\frac{1}{2}$. At the same time, $\frac{1}{2}$ is the *largest upper area bound* one can hope for, because two squares larger than the quarter-squares shown in Fig. 1 cannot be packed. We call the ratio between the largest combined object area that can always be packed and the area of the container the problem's *critical density*, or *worst-case density*. The equivalent problem of establishing the critical packing density for disks in a square was posed by Demaine, Fekete, and Lang [2] and resolved by Morr, Fekete and Scheffer [17, 4]. Making use of a recursive procedure for cutting the container into triangular pieces, they proved that the critical packing density of disks in a square is $\frac{\pi}{3+2\sqrt{2}} \approx 0.539$. It is quite natural to consider the analogous question of establishing the critical packing density for disks in a disk. However, the shelf-packing approach of Moon and Moser [16] uses the fact that rectangular shapes of the packed objects fit well into parallel shelves, which is not the case for disks; on the other hand, the split packing method of Morr et al. [17, 4] relies on recursively splitting triangular containers, so it does not work for a circular container that cannot be partitioned into smaller circular pieces.

1.1 Results

We prove that the critical density for packing disks into a disk is 1/2: Any set of not necessarily equal disks with a combined area of not more than half the area of a circular container can be packed; this is best possibly, as for any $\varepsilon > 0$ there are instances of total area $1/2 + \varepsilon$ that cannot be packed. See Fig. 2 for the critical configuration.

Our proofs are constructive, so they can also be used as a constant-factor approximation algorithm for the smallest-area container of a given shape in which a given set of disks can be packed. Due to the higher geometric difficulty of fitting together circular objects, the involved methods are considerably more complex than those for square containers. We make up for this difficulty by developing more intricate recursive arguments, including appropriate and powerful tools based on *interval arithmetic*.

2 A Worst-Case Optimal Algorithm

▶ **Theorem 1.** *Every set of disks with total area $\frac{\pi}{2}$ can be packed into the unit disk O with radius 1. For any $\varepsilon > 0$, there is a set of disks with total area $\frac{\pi}{2} + \varepsilon$ that cannot be packed into O. In other words, the* worst-case packing density *for packing disks into a disk is $\frac{1}{2}$.*

The worst case consists of two disks D_1, D_2 with radius $\frac{1}{2}$, see Fig. 2. Increasing the area of D_1 by ε yields a set of disks which cannot be packed. The total area of these two disks is

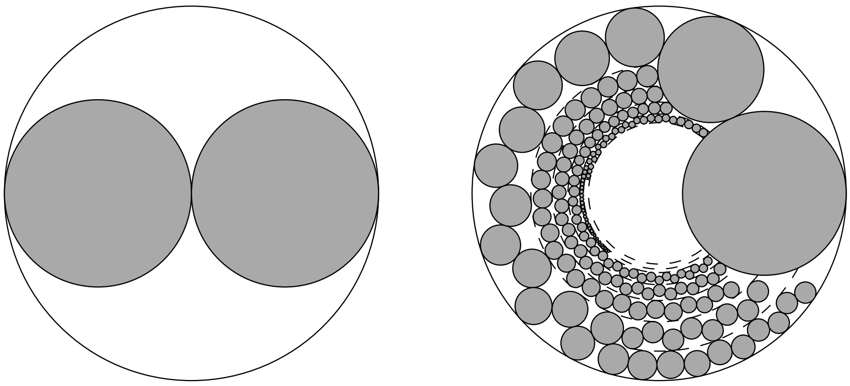

Figure 2 (1) A critical instance that allows a packing density no better than $\frac{1}{2}$. (2) An example packing produced by our algorithm.

$\frac{\pi}{4} + \frac{\pi}{4} = \frac{\pi}{2}$.

In the remainder of Section 2, we give a constructive proof for Theorem 1. Before we proceed to describe our algorithm in Section 2.4, we give some definitions and describe *Disk Packing* and *Ring Packing* as two subroutines of our algorithm.

2.1 Preliminaries for the Algorithm

We make use of the following definitions, see Fig. 3.

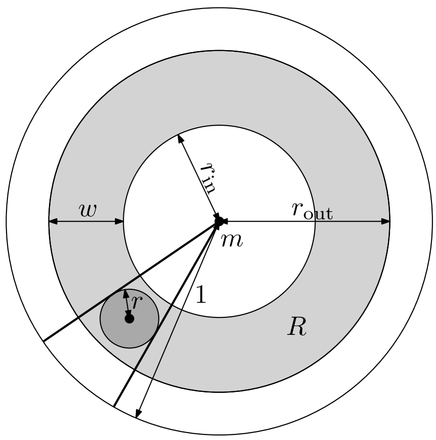

Figure 3 A ring $R \subset O$ with width w and a disk with its corresponding tangents.

For $r_{\text{out}} > r_{\text{in}} \geq 0$ and a container disk $\mathcal{C}$ such that $r_{\text{out}} \leq 2r_{\text{in}}$, we define a *ring* $R := R[r_{\text{out}}, r_{\text{in}}]$ of $\mathcal{C}$ as the closure of $r_{\text{out}} \setminus r_{\text{in}}$, see Fig. 3. If $r_{\text{in}} > 0$, the boundary of R consists of two connected components. The *inner boundary* is the component that lies closer to the center m of $\mathcal{C}$ and the *outer boundary* is the other component. The *inner radius* and the *outer radius* of R are the radius of the inner boundary and the radius of outer boundary. Each ring considered by our algorithm has one of three states $\{\text{OPEN}, \text{CLOSED}, \text{FULL}\}$. Initially, after its construction by the algorithm, each ring is OPEN.

Let r be a disk inside a container disk $\mathcal{C}$. The two *tangents* of r are the two rays starting in the center of $\mathcal{C}$ and touching the boundary of r. We say that a disk lies *adjacent* to r_{out} when the disk is touching the boundary of r_{out} from the inside of r_{out}.

2.2 Disk Packing: A Subroutine

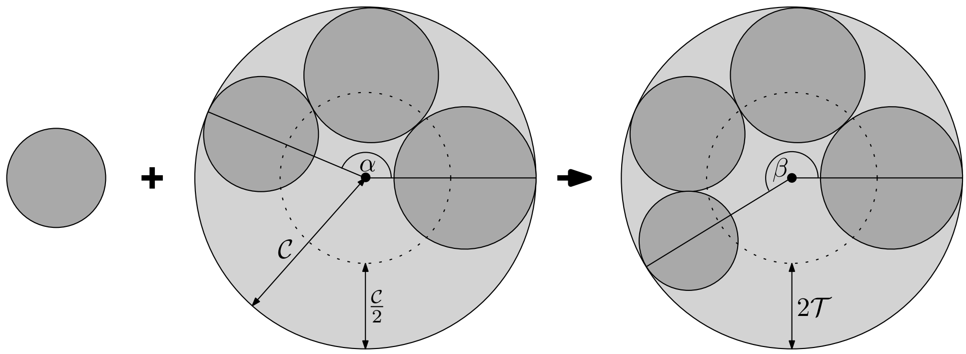

Figure 4 Disk Packing places disks in decreasing order of radius into a container $\mathcal{C}$ adjacent to the boundary of $\mathcal{C}$.

Consider a container disk $\mathcal{C}$, a set S of already packed disks that overlap with $\mathcal{C}$, but are not necessarily contained in it, and another disk r_i to be packed; see Fig. 4. We *pack* r_i into $\mathcal{C}$ adjacently to the boundary of $\mathcal{C}$ as follows: Let α be the maximal polar angle realized by the center of any disk from S. We choose the center of r_i such that it realizes the smallest possible polar angle $\beta \geq \alpha$ such that r_i touches the outer boundary of $\mathcal{C}$ from the interior of $\mathcal{C}$ without overlapping another disk from S, see Fig. 4. If r_i cannot be packed into $\mathcal{C}$, we say that r_i *does not fit into* R.

Let $0 < \mathcal{T} \leq \frac{1}{4}$, called *threshold*. *Disk Packing* considers the disks in decreasing order of radius and packs each disk r_i adjacent to the previous disk r_{i-1} and the boundary of $\mathcal{C}$ until r_i does not fit into $\mathcal{C}$ or $r_i < \mathcal{T}$.

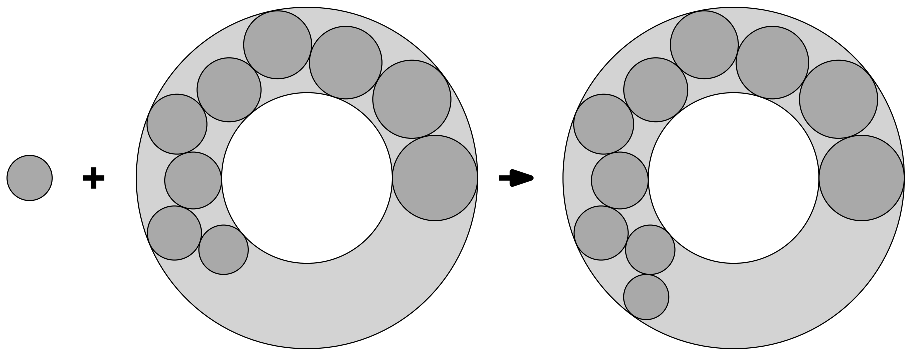

Figure 5 Ring Packing packs disks into a ring $R[r_{\text{out}}, r_{\text{in}}]$, alternating adjacent to the outer and to the inner boundary of R.

2.3 Ring Packing: A Subroutine

Consider a ring $R := R[r_{\text{out}}, r_{\text{in}}]$ with inner radius r_{in} and outer radius r_{out}, a (possibly empty) set S of already packed disks that overlap with R, and another disk r_i to be packed, see Fig. 5. We *pack* r_i into R adjacent to the outer (inner) boundary of R as follows: Let α be the maximal polar angle realized by a midpoint of a disk from S. We choose the midpoint of r_i realizing a smallest possible polar angle $\beta \geq \alpha$ such that r_i touches the outer (inner) boundary of R from the interior of R without overlapping another disk from S. If r_i cannot be packed into R, we say that r_i *does not fit into R (adjacent to the outer (inner) boundary)*.

Ring Packing iteratively packs disks into R alternating adjacent to the inner and outer boundary. If the current disk r_i does not fit into R, Ring Packing stops and we declare R to be FULL. If r_{i-1} and r_i could pass each other in R, i.e., the sum of the diameters of r_{i-1} and r_i are smaller than the width of R, Ring Packing stops and we declare R to be CLOSED.

2.4 Description of the Algorithm

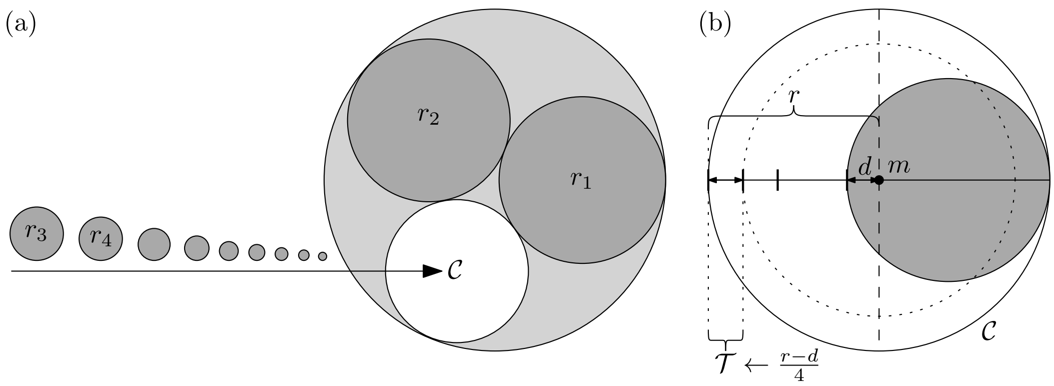

Figure 6 (a): If $r_1, r_2 \geq 0.495\mathcal{C}$, Disk Packing packs r_1, r_2 into $\mathcal{C}$. We update the current container disk $\mathcal{C}$ as the largest disk that fits into $\mathcal{C}$ and recurse on $\mathcal{C}$ with $r_3, \dots, r_n$. (b): Determining the threshold $\mathcal{T}$ for disks packed by Disk Packing.

Our algorithm *creates* rings. A ring only exists after it is created. We stop packing at any point in time when all disks are packed. Furthermore, we store the current threshold $\mathcal{T}$ for Disk Packing and the smallest inner radius $r_{\min}$ of a ring created during the entire run of our algorithm. Initially, we set $\mathcal{T} \leftarrow 1, r_{\min} \leftarrow 1$. Our algorithm works in five phases:

- **Phase 1 - Recursion:** If $r_1, r_2 \geq 0.495\mathcal{C}$, apply Disk Packing to r_1, r_2, update $\mathcal{C}$ as the largest disk that fits into $\mathcal{C}$ and $\mathcal{T}$ as the radius of $\mathcal{C}$, and recurse on $\mathcal{C}$, see Fig. 6(a).
- **Phase 2 - Disk Packing:** Let r be the radius of $\mathcal{C}$. If the midpoint m of $\mathcal{C}$ lies inside a packed disk r_i, let d be the minimal distance of m to the boundary of r_i, see Fig. 6(b). Otherwise, we set $d = 0$.
  We apply Disk Packing to the container disk $\mathcal{C}$ with the threshold $\mathcal{T} \leftarrow \frac{r-d}{4}$.
- **Phase 3 - Ring Packing:** We apply Ring Packing to the ring $R := R[r_{\text{out}}, r_{\text{in}}]$ determined as follows: Let r_i be the largest disk not yet packed. If there is no open ring inside $\mathcal{C}$, we create a new open ring $R[r_{\text{out}}, r_{\text{in}}] \leftarrow R[r_{\min}, r_{\min} - 2r_i]$. Else, let $R[r_{\text{out}}, r_{\text{in}}]$ be the open ring with the largest inner radius r_{in}.
- **Phase 4 - Managing Rings:** Let $R[r_{\text{out}}, r_{\text{in}}]$ be the ring filled in Phase 3. We declare $R[r_{\text{out}}, r_{\text{in}}]$ to be closed and proceed as follows: Let r_i be the largest disk not yet packed.

If r_i and r_{i+1} can pass one another inside $R[r_{\text{out}}, r_{\text{in}}]$, i.e., if $2r_i + 2r_{i+1} \leq r_{\text{out}} - r_{\text{in}}$, we create two new open rings $R[r_{\text{out}}, r_{\text{out}} - 2r_i]$ and $R[r_{\text{out}} - 2r_i, r_{\text{in}}]$.

- **Phase 5 - Continue:** If there is an open ring, we go to Phase 3. Otherwise, we set $\mathcal{C}$ as the largest disk not covered by created rings, set $\mathcal{T}$ as the radius of $\mathcal{C}$, and go to Phase 2.

3 Analysis of the Algorithm

The analysis uses an intricate combination of manual analysis and an automated analysis based on interval arithmetic. For lack of space, details are omitted. See the appendix for full details.

4 Hardness

It is straightforward to see that the hardness proof for packing disks into a square can be adapted to packing disks into a disk, as follows.

▶ **Theorem 2.** *It is NP-hard to decide whether a given set of disks fits into a circular container.*

The proof is completely analogous to the one by Demaine, Fekete, and Lang in 2010 [2], who used a reduction from 3-Partition. Their proof constructs a disk instance which first forces some symmetrical free "pockets" in the resulting disk packing. The instance's remaining disks can then be packed into these pockets if and only if the related 3-Partition instance has a solution. Similar to their construction, we construct a symmetric triangular pocket by using a set of three identical disks of radius $\frac{\sqrt{3}}{2+\sqrt{3}}$ that can only be packed into a unit disk by touching each other. Analogous to [2], this is further subdivided into a sufficiently large set of identical pockets. The remaining disks encode a 3-Partition instance that can be solved if and only if the disks can be partitioned into triples of disks that fit into these pockets.

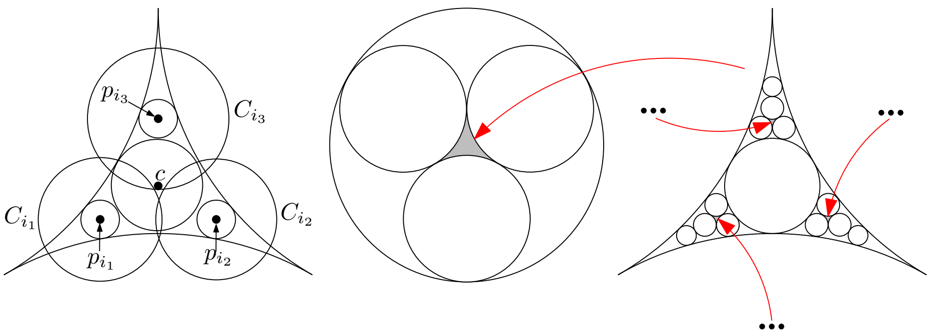

Figure 7 Elements of the hardness proof: (1) A symmetric triangular pocket from [2], allowing three disks with centers p_{i_1}, p_{i_2}, p_{i_3} to be packed if and only if the sum of the three corresponding numbers from the 3-Partition instance is small enough. (2) Creating a symmetric triangular pocket in the center by packing three disks of radius $\frac{\sqrt{3}}{2+\sqrt{3}}$ and the adapted argument from [2] for creating a sufficiently large set of symmetric triangular pockets.

5 Conclusions

We have established the critical density for packing disks into a disk, based on a number of advanced techniques that are more involved than the ones used for packing squares into a square or disks into a square. Numerous questions remain, in particular the critical density for packing disks of bounded size into a disk or the critical density of packing squares into a disk. These remain for future work; we are optimistic that some of our techniques will be useful.

References

1 I. Castillo, F. J. Kampas, and J. D. Pintér. Solving circle packing problems by global optimization: numerical results and industrial applications. *European Journal of Operational Research*, 191(3):786–802, 2008.

2 E. D. Demaine, S. P. Fekete, and R. J. Lang. Circle packing for origami design is hard. In *Origami^5: 5th International Conference on Origami in Science, Mathematics and Education*, AK Peters/CRC Press, pages 609–626, 2011.

3 S. P. Fekete, P. Keldenich, and C. Scheffer. Packing disks into disks with optimal worst-case density. In *Proceedings of the 35th Symposium on Computational Geometry*, 2019. To appear.

4 S. P. Fekete, S. Morr, and C. Scheffer. Split packing: Algorithms for packing circles with optimal worst-case density. *Discrete & Computational Geometry*, 2018.

5 F. Fodor. The densest packing of 19 congruent circles in a circle. *Geometriae Dedicata*, 74:139–145, 1999.

6 F. Fodor. The densest packing of 12 congruent circles in a circle. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 41:401–409, 2000.

7 F. Fodor. The densest packing of 13 congruent circles in a circle. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 44:431–440, 2003.

8 H. J. Fraser and J. A. George. Integrated container loading software for pulp and paper industry. *European Journal of Operational Research*, 77(3):466–474, 1994.

9 M. Goldberg. Packing of 14, 16, 17 and 20 circles in a circle. *Mathematics Magazine*, 44:134–139, 1971.

10 R. Graham, B. Lubachevsky, K. Nurmela, and P. Östergøard. Dense packings of congruent circles in a circle. *Discrete Mathematics*, 181:139–154, 1998.

11 M. Hifi and R. M'hallah. A literature review on circle and sphere packing problems: models and methodologies. *Advances in Operations Research*, 2009. Article ID 150624.

12 S. Kravitz. Packing cylinders into cylindrical containers. *Mathematics Magazine*, 40:65–71, 1967.

13 R. J. Lang. A computational algorithm for origami design. *Proceedings of the Twelfth Annual Symposium on Computational Geometry (SoCG)*, pages 98–105, 1996.

14 B. Lubachevsky and R. Graham. Curved hexagonal packings of equal disks in a circle. *Discrete & Computational Geometry*, 18:179–194, 1997.

15 H. Melissen. Densest packing of eleven congruent circles in a circle. *Geometriae Dedicata*, 50:15–25, 1994.

16 J. W. Moon and L. Moser. Some packing and covering theorems. In *Colloquium Mathematicae*, volume 17, pages 103–110. Institute of Mathematics, Polish Academy of Sciences, 1967.

17 S. Morr. Split packing: An algorithm for packing circles with optimal worst-case density. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 99–109, 2017.

18 N. Oler. A finite packing problem. *Canadian Mathematical Bulletin*, 4:153–155, 1961.

19 R. Peikert, D. Würtz, M. Monagan, and C. de Groot. Packing circles in a square: A review and new results. In *Proceedings of the 15th IFIP Conference*, pages 45–54, 1992.

20 G. Reis. Dense packing of equal circles within a circle. *Mathematics Magazine*, issue 48:33–37, 1975.

21 E. Specht. Packomania, 2015. http://www.packomania.com/.

22 K. Sugihara, M. Sawai, H. Sano, D.-S. Kim, and D. Kim. Disk packing for the estimation of the size of a wire bundle. *Japan Journal of Industrial and Applied Mathematics*, 21(3):259–278, 2004.

23 P. G. Szabó, M. C. Markót, T. Csendes, E. Specht, L. G. Casado, and I. García. *New Approaches to Circle Packing in a Square*. Springer US, 2007.

24 H. Wang, W. Huang, Q. Zhang, and D. Xu. An improved algorithm for the packing of unequal circles within a larger containing circle. *European Journal of Operational Research*, 141(2):440–453, sep 2002.

25 D. Würtz, M. Monagan, and R. Peikert. The history of packing circles in a square. *Maple Technical Newsletter*, page 35–42, 1994.