Split Packing: Packing Circles into Triangles with Optimal Worst-Case Density

Sándor P. Fekete, Sebastian Morr, and Christian Scheffer

Department of Computer Science, TU Braunschweig, Germany
s.fekete@tu-bs.de, sebastian@morr.cc, scheffer@ibr.cs.tu-bs.de

Abstract. In the *circle packing problem for triangular containers*, one asks whether a given set of circles can be packed into a given triangle. Packing problems like this have been shown to be NP-hard. In this paper, we present a new sufficient condition for packing circles into any right or obtuse triangle using only the circles' combined area: It is possible to pack any circle instance whose combined area does not exceed the triangle's incircle. This area condition is tight, in the sense that for any larger area, there are instances which cannot be packed.
A similar result for square containers has been established earlier this year, using the versatile, divide-and-conquer-based Split Packing algorithm. In this paper, we present a generalized, weighted version of this approach, allowing us to construct packings of circles into asymmetric triangles. It seems crucial to the success of these results that Split Packing does not depend on an orthogonal subdivision structure. Beside realizing all packings below the critical density bound, our algorithm can also be used as a constant-factor approximation algorithm when looking for the smallest non-acute triangle of a given side ratio in which a given set of circles can be packed.
An interactive visualization of the Split Packing approach and other related material can be found at https://morr.cc/split-packing/.

1 Introduction

Given a set of circles, can you decide whether it is possible to pack these circles into a given container without overlapping one another or the container's boundary? This naturally occurring *circle packing problem* has numerous applications in engineering, science, operational research and everyday life. Examples include packaging cylinders [2], bundling tubes or cables [16, 18], the cutting industry [17], the layout of control panels [2], the design of digital modulation schemes [14], or radio tower placement [17]. Further applications stem from chemistry [19], foresting [17], and origami design [9].

Despite its simple formulation, packing problems like these were shown to be NP-hard in 2010 by Demaine, Fekete, and Lang [3], using a reduction from 3-Partition. Additionally, due to the irrational coordinates which arise when packing circular objects, it is also surprisingly hard to solve circle packing problems in practice. Even when the input consists of equally-sized circles, exact

F. Ellen et al. (Eds.): WADS 2017, LNCS 10389, pp. 373–384, 2017.
DOI: 10.1007/978-3-319-62127-2_32

boundaries for the smallest square container are currently only known for up to 35 circles, see [10]. For right isosceles triangular containers, optimal results have been published for up to 7 equal circles, see [20].

The related problem of packing square objects has long been studied. Already in 1967, Moon and Moser [12] found a sufficient condition: They proved that it is possible to pack a set of squares into the unit square in a shelf-like manner if their combined area does not exceed $1/2$, see Figure 2. At the same time, $1/2$ is the *largest upper area bound* you could hope for, because two squares larger than the quarter-squares depicted in Figure 1 cannot be packed anymore. 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.*

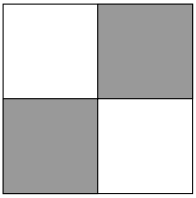

Fig. 1. Worst-case instance for packing squares into a square.

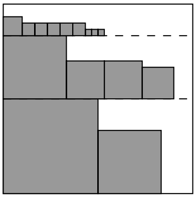

Fig. 2. Example of Moon and Moser's shelf-packing.

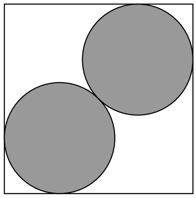

Fig. 3. Worst-case instance for packing circles into a square.

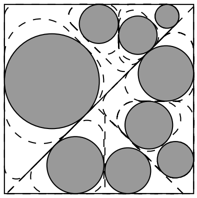

Fig. 4. Example packing produced by Split Packing.

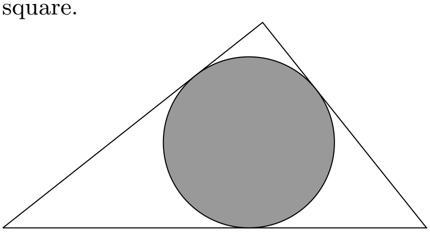

Fig. 5. Suspected worst-case instance for packing circles into a non-acute triangle.

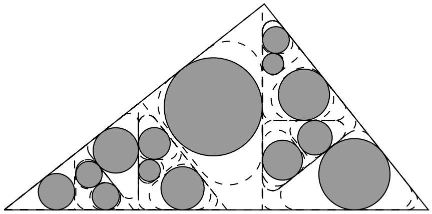

Fig. 6. Example packing produced by Split Packing.

We recently showed a similar result for circular objects: Each circle instance not exceeding the area of the instance shown in Figure 3 can be packed, and this area condition is tight [13]. Proving this required a fundamentally different approach than Moon and Moser's orthogonal shelf-packing, compare Figure 4.

In this paper, we consider the problem of packing circles into non-acute triangular containers. It is obvious that circles larger than a triangle's incircle cannot be packed (compare Figure 5), but is it also possible to pack all circle instances of up to that combined area? We will answer this question affirmatively and introduce a weighted modification of the Split Packing algorithm, allowing us to pack circles into asymmetric non-acute triangles with critical density. See Figure 6 for an example packing.

Many authors have considered heuristics for circle packing problems, see [7, 17] for overviews of numerous heuristics and optimization methods. The best known solutions for packing equal circles into squares, triangles and other shapes are continuously published on Specht's website http://packomania.com [15].

That being said, the literature on exact approximation algorithms which actually give performance guarantees is small. Miyazawa et al. [11] devised asymptotic polynomial-time approximation schemes for packing circles into the smallest number of unit square bins. And recently, Hokama, Miyazawa, and Schouery [8] developed an asymptotic approximation algorithm for the online version of that problem. To the best of our knowledge, this paper presents the first approximation algorithm for packing circles into triangular containers.

1.1 Results

We show that, for any right or obtuse triangle, any circle instance with a combined area of up to the triangle's incircle can be packed into that triangle. At the same time, for any larger area, there are instances which cannot be packed, making the ratio between the incircle's and the triangle's area the triangle's *critical density*. For a right isosceles triangle, this density is approximately 53.91%. In the general case, the critical density of a non-acute triangle with side lengths a, b, and c is

$$\sqrt{\frac{-(a-b-c)(a+b-c)(a-b+c)}{(a+b+c)^3}}\pi.$$

Our proof is constructive: The *Split Packing algorithm* can be used to construct the packings in polynomial time. Split Packing can also be used as a constant-factor approximation algorithm of the smallest-area non-acute triangle of a given side ratio which can pack a given set of circles. The approximation factor is the reciprocal of the critical density.

While we focus on triangular containers in this paper, we see more opportunities to generalize the Split Packing approach for other container and object types. We discuss some of these extensions in the conclusion on page 11.

1.2 Key ideas

The *Split Packing* approach, which we successfully used for packing circles into square containers earlier this year [13], is built on two basic ideas:

First, it applies a recursive subdivision strategy, which cuts the container into smaller triangles, while keeping the combined area of the triangles' incircles constant. And second, it performs the splitting of the circle instance into subgroups using an algorithm which resembles greedy scheduling. This makes sure the resulting subgroups are *close* to equal in terms of their combined area. If the groups' areas deviate from the targeted 1:1 ratio, we can gain information about the minimum circle size in the larger group, allowing us to round off the subcontainer triangles.

In this paper, we introduce a weighted generalization of the Split Packing approach: When packing into asymmetric triangles, we do not want the resulting groups to have equal area, as it is not possible to cut the container into two subtriangles of equal size. Instead, we target a different area ratio, defined by the incircles of the two triangles created by cutting the container orthogonally its the base through its tip, see Figure 8 on page 8. We call this desired area ratio the *split key*.

The rest of the paper will detail this process.

2 Greedy splitting

The following definitions makes it easier to talk about the properties of circle instances:

Definition 1. *A* circle instance *is a multiset of nonnegative real numbers, which define the circles' areas. For any circle instance C,* $\text{sum}(C)$ *is the combined area of the instance's circles and* $\min(C)$ *is the area of the smallest circle contained in the instance.*

Definition 2. $\mathbb{C}$ *is the set of all circle instances.* $\mathbb{C}(a)$ *consists of exactly those circle instances C with* $\text{sum}(C) \leq a$. *Finally,* $\mathbb{C}(a, b)$ *consists of exactly those circle instances $C \in \mathbb{C}(a)$ with* $\min(C) \geq b$.

Algorithm 1 takes a circle instance C, and splits it into two groups according to the *split key* F, which determines the targeted ratio of the resulting groups' combined areas. Because the method resembles a greedy scheduling algorithm, we call the process *greedy splitting*. The algorithm first creates two empty "buckets", and in each step adds the largest remaining circle of the input instance to the "relatively more empty" bucket:

Algorithm 1 $\textsc{Split}(C, F)$

Input: A circle instance C, sorted by size in descending order, and a split key $F = (f_1, f_2)$
Output: Circle instances C_1, C_2

```
C_1 ← ∅
C_2 ← ∅
for all c ∈ C do
    j = arg min_i sum(C_i)/f_i        ▷ Find the index of the more empty bucket.
    C_j ← C_j ∪ {c}
end for
```

If the resulting groups' area ratio deviates from the area ratio targeted by the split key, we gain additional information about the "relatively larger" group:

The more this group exceeds its targeted ratio, the larger the minimum size of its elements, allowing a "more rounded" subcontainer in the packing. See Figure 9 on page 9 for an illustration.

Lemma 1. *For any C_1 and C_2 produced by* SPLIT$(C, (f_1, f_2))$*:*

$$\min(C_i) \geq \text{sum}(C_i) - f_i \frac{\text{sum}(C_j)}{f_j}$$

Proof. If $\frac{\text{sum}(C_i)}{f_i} < \frac{\text{sum}(C_j)}{f_j}$, then the lemma says that $\min(C_i)$ is larger than a negative number, which is certainly true.

Otherwise, set $r := \frac{\text{sum}(C_j)}{f_j}$. This value describes the smaller "relative filling level" by the time the algorithm ends. Now assume for contradiction C_i contained an element smaller than $\text{sum}(C_i) - f_i r$. As the elements were inserted by descending size, all elements which were put into C_i after that element would have to be at least as small. So the final element put into C_i (let us call it c) would be smaller than $\text{sum}(C_i) - f_i r$, as well.

But this means that

$$\frac{\text{sum}(C_i) - c}{f_i} > \frac{\text{sum}(C_i) - (\text{sum}(C_i) - f_i r)}{f_i} = r,$$

meaning that at the moment before c was inserted, the relative filling level of C_i would already have been larger than r. Recall that r is the smallest filling level of any group by the time the algorithm ends, meaning that at the time when c is inserted, C_i's filling level is already larger than the filling level of the other group. This is a contradiction, as the greedy algorithm would choose to put c not into C_i, but into the other group with the smaller filling level in this case. □

We are now going to define a term which encapsulates all properties of the circle instances output by SPLIT. These properties depend on the used split key F, and also on the combined area a and the minimum circle size b of the circle instance, which is why it the term has three parameters.

Definition 3. *For any $0 \leq b \leq a$ and any split key $F = (f_1, f_2)$, we say that the tuples (a_1, b_1), (a_2, b_2) are (a, b, F)-*conjugated *if*

- $a_1 + a_2 = a$,
- $b_i \geq b$, *and*
- $b_i \geq a_i - f_i \frac{a_j}{f_j}$.

Two circle instances C_1 and C_2 are (a, b, F)-conjugated if there are any (a, b, F)-conjugated tuples (a_1, b_1) and (a_2, b_2) so that $C_1 \in \mathbb{C}(a_1, b_1)$ and $C_2 \in \mathbb{C}(a_2, b_2)$.

We can now associate this property with SPLIT in the following theorem:

Theorem 1. *For any $C \in \mathbb{C}(a, b)$ and any split key $F = (f_1, f_2)$,* SPLIT(C, F) *always produces two (a, b, F)-conjugated subinstances.*

Proof. That the subinstances' combined areas add up to a follows directly from the algorithm. As the minimum size of all circles in C is b, this must also be true for the subinstances, so $\min(C_i) \geq b$. The other minimum-size property follows from Lemma 1. □

3 Split Packing

The SPLIT algorithm presented in the previous section, in addition to the properties of the instances it produces, are the foundations on which we now build the central theorem of this paper. Split Packing by itself is a general framework to pack circles and other shapes into containers. We will apply the Split Packing theorem to triangular containers in the next section.

We will often want to state that a shape can pack all circle instances which belong to a certain class. For this, we define the term $\mathcal{C}$*-shape*:

Definition 4. *For any $\mathcal{C} \subseteq \mathbb{C}$, a $\mathcal{C}$-*shape *is a shape in which each $C \in \mathcal{C}$ can be packed.*

For example, if a shape is a $\mathbb{C}(a)$-shape, it means that it can pack all circle instances with a combined area of a. And a $\mathbb{C}(a, b)$-shape can pack all circle instances with a combined area of a, whose circles each have an area of at most b.

We can now state our central theorem: If it is possible to find two subcontainers which fit in a given shape, and which can pack all possible subinstances produced by SPLIT, it is possible to pack the original class of circle instances into that shape.

Theorem 2 (Split Packing). *A shape s is a $\mathbb{C}(a, b)$-shape if there is a split key F, so that for all (a, b, F)-conjugated tuples (a_1, b_1) and (a_2, b_2) one can find a $\mathbb{C}(a_1, b_1)$-shape and a $\mathbb{C}(a_2, b_2)$-shape which can be packed into s.*

Proof. Consider an arbitrary $C \in \mathbb{C}(a, b)$. We use SPLIT(C, F) to produce two subinstances C_1 and C_2. We know from Theorem 1 that those subinstances will always be (a, b, F)-conjugated. So if we can indeed find two shapes which can pack these subinstances, and if we can pack these two shapes into s, then we also can pack the original circle instance C into s.

Note that in the special case that C consists of a single circle, SPLIT(C, F) will yield two circle instances $C_1 = \{C\}$ and $C_2 = \emptyset$. For this case, Theorem 1 guarantees a minimum size of a for the first group, and the associated $\mathbb{C}(a_1, b_1)$-shape is just an a-circle. This means that we can simply place the input circle in the container, and stop the recursion at this point. □

Written as an algorithm, Split Packing looks like this:

Algorithm 2 SPLITPACK(s, C)

Input: A $\mathbb{C}(a,b)$-shape s and a circle instance $C \in \mathbb{C}(a,b)$, sorted by size in descending order
Output: A packing of C into s

```
Determine split key F for shape s
(C_1, C_2) ← SPLIT(C, F)                          ▷ See Algorithm 1.
for all i ∈ {1, 2} do
    a_i ← sum(C_i)
    b_i ← minimum guarantee for C_i               ▷ See Lemma 1.
    Determine a ℂ(a_i, b_i)-shape s_i
    SPLITPACK(s_i, C_i)
end for
Pack s_1, s_2, and their contents into s
```

Note that the Split Packing algorithm can easily be extended to allow splitting into more than two subgroups. For simplicity, we only describe the case of two subgroups here, as this suffices for the shapes we discuss in this paper.

3.1 Analysis

The analysis of the Split Packing approach follows exactly the same lines as in our previous paper [13]. We will repeat the result here without proof.

Theorem 3. *Split Packing requires $\mathcal{O}(n)$ basic geometric constructions and $\mathcal{O}(n^2)$ numerical operations.*

Theorem 4. *Split Packing, when used to pack circles into a $\mathbb{C}(a,b)$-shape of area A, is an approximation algorithm with an approximation factor of $\frac{A}{a}$, compared to the container of minimum area.*

4 Packing into hats

After this general description of Split Packing, we will now apply it to concrete containers. We start with an observation:

If all circles which we want to pack have a certain minimum size, sharp corners of the container cannot be utilized anyway. This observation motivates a family of shapes which resemble rounded triangles. We call these shapes *hats*:

Definition 5. *For each $0 \leq b \leq a$, an (a,b)-*hat *is a non-acute triangle with an incircle of area a, whose corners are rounded to the radius of a b-circle, see Figure 7. Call the two smaller angles of the original triangle* left-angle *and* right-angle*. If we say* right hat*, the hat is based on a right triangle.*

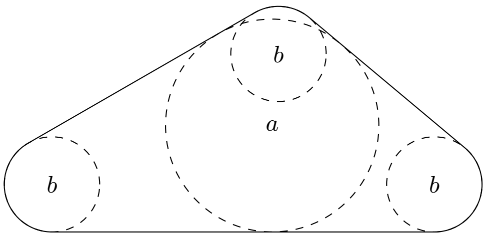

Fig. 7. An (a, b)-hat.

We will now proceed to show that all circle instances with a combined area of up to a with a minimum circle size of b can be packed into an (a, b)-hat.

First, it is important to choose the correct split key when packing into asymmetric hats. We are aiming for a group ratio which will lead to a cut through the hat's tip if it is reached exactly:

Definition 6. *To get a hat's* associated split key*, split the underlying triangle orthogonally to its base through its tip, and inscribe two circles in the two sides, see Figure 8. The areas of these circles are the two components of the hat's split key.*

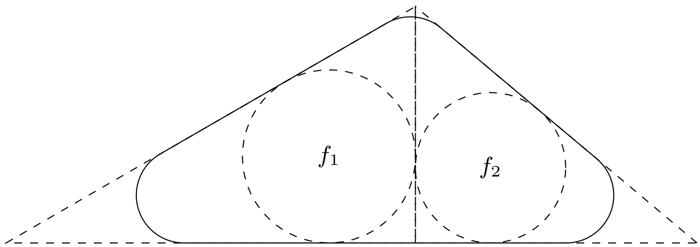

Fig. 8. A hat's *associated split key* equals (f_1, f_2)

Lemma 2. *Consider an $(a, 0)$-hat with the associated split key $F = (f_1, f_2)$, and call its left- and right-angles α and β. For all $(a, 0, F)$-conjugated tuples (a_1, b_1) and (a_2, b_2), the following two shapes can be packed into the hat:*

- *a right (a_1, b_1)-hat with a right-angle of α and*
- *a right (a_2, b_2)-hat with a left-angle of β.*

The proof of this theorem is rather technical in nature. We omit it here due to space constraints, refer to the full version [6]. See Figure 9 for an intuition of what the resulting hats look like. Note that, as the hats' incircles are getting larger than the targeted area ratio, their corners become more rounded so that they don't overlap the container's boundary.

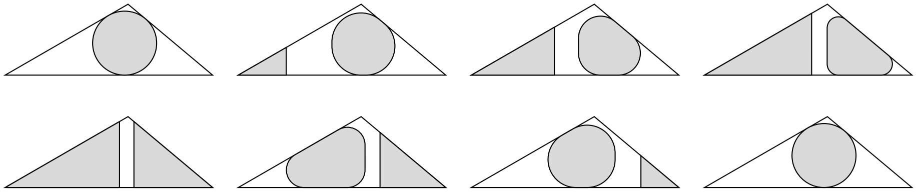

Fig. 9. Hat-in-hat packings for different ratios of a_1 and a_2.

In the previous lemma, the container is always an $(a, 0)$-hat, which is essentially a non-rounded triangle with an incircle of a. The next lemma extends this idea to hats which are actually rounded. It is identical to Lemma 2, except that the rounding of the container hat is no longer 0, but b.

Lemma 3. *Consider an (a, b)-hat with the associated split key $F = (f_1, f_2)$, and call its left- and right-angles α and β. For all (a, b, F)-conjugated tuples (a_1, b_1) and (a_2, b_2) with $a_1 + a_2 \leq a$, the following two shapes can be packed into the hat:*

- *a right (a_1, b_1)-hat with a right-angle of α and*
- *a right (a_2, b_2)-hat with a left-angle of β.*

Proof. Lemma 2 tells us that this theorem is true for $b = 0$. Now, the container's corners can be rounded to the radius of a b-circle, and we need to show that the two hats from the previous construction still fit inside. But all of the two hat's corners are also rounded to (at least) the same radius (see Theorem 1), so they will never overlap the container, see Figure 10. □

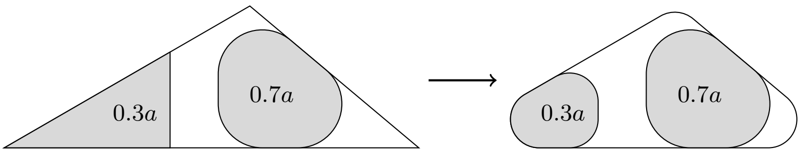

Fig. 10. Rounding all hats' corners by the same radius does not affect the packing.

With these preparations, we can apply Split Packing to hats:

Theorem 5. *Given an (a, b)-hat, all circle instances with a combined area of at most a and a minimum circle size of at least b can be packed into that hat.*

Proof. We proof by induction that we can pack each $C \in \mathbb{C}(a, b)$ into the hat:

If C only consists of a single circle, it can be packed into the hat, as it is at most as big as the hat's incircle.

Now assume that for any $0 \leq b \leq a$, any (a, b)-hat could pack all circle instances into $\mathbb{C}(a, b)$ with at most n circles. Consider a circle instance $C \in \mathbb{C}(a, b)$

containing $n+1$ circles. Definition 6 tells us how to compute the split key F. Then we know from Theorem 1 that SPLIT will partition C into two subinstances $C_1 \in \mathbb{C}(a_1, b_1)$ and $C_2 \in \mathbb{C}(a_2, b_2)$, whose parameters are (a, b, F)-conjugated. As SPLIT can never return an empty instance (except for $|C| = 1$, a case which we handled above), each subinstance will contain at most n circles. We know from Lemma 3 that, for all pairs of (a, b, F)-conjugated tuples, we can find two hats with matching parameters which fit into the container hat. By assumption, these hats can now pack all instances from $\mathbb{C}(a_1, b_1)$ and $\mathbb{C}(a_2, b_2)$, respectively, which means that they can especially also pack C_1 and C_2. If we then pack the two hats into the container, we have constructed a packing of C into the container hat.

By induction, we can pack each $C \in \mathbb{C}(a, b)$ into the (a, b)-hat. □

Finally, we can state this paper's central result:

Theorem 6. *Given a non-acute triangle with an incircle of area a, all circle instances with a combined area of up to a can be packed into the triangle, and this bound is tight. Expressed algebraically, for a triangle with side lengths a, b, and c, the critical density is*

$$\sqrt{\frac{-(a-b-c)(a+b-c)(a-b+c)}{(a+b+c)^3}}\pi.$$

Proof. The triangle is an $(a, 0)$-hat, which by Theorem 5 is a $\mathbb{C}(a)$-shape.

On the other hand, a single circle of area $a + \varepsilon$ cannot be packed, as the incircle is by definition the largest circle which fits into the triangle.

As for the algebraic formulation of the critical density, the area of the triangle can be calculated using Heron's formula:

$$\Delta(a, b, c) := \sqrt{s(s-a)(s-b)(s-c)} \text{ with } s = \frac{a+b+c}{2}$$

It is also known that the radius of the incircle of this triangle is

$$R(a, b, c) := \frac{\Delta(a, b, c)}{s} \text{ with } s = \frac{a+b+c}{2},$$

so the incircle has an area of

$$I(a, b, c) = \pi R(a, b, c)^2 = \frac{(a+b-c)(c+a-b)(b+c-a)}{4(a+b+c)}.$$

Finally, the ratio between the areas of the circle and the triangle can be calculated to be

$$\frac{I(a, b, c)}{\Delta(a, b, c)} = \sqrt{\frac{-(a-b-c)(a+b-c)(a-b+c)}{(a+b+c)^3}}\pi,$$

For a right isosceles triangle, this density is approximately 53.91%. □

5 Conclusion

In this paper, we presented a constructive proof of the critical densities when packing circles into right or obtuse triangles, using a weighted Split Packing technique. We see more opportunities to apply this approach in the context of other packing and covering problems.

It is possible to use Split Packing to pack into **more container types**. At this point, we can establish the critical densities for packing circles into equilateral triangles and rectangles exceeding a certain aspect ratio. The case of acute triangles is still open, we discuss why the approach presented in this paper does not work there in the full version [6].

Split Packing can also be extended to pack **objects other than circles**. We can establish the critical densities for packing octagons into squares, and think we can describe the maximum shape which can be packed into squares using Split Packing.

Another natural extension is the **online version** of the problem. The current best algorithm that packs squares into a square in an online fashion by Brubach [1], based on the work by Fekete and Hoffmann [4, 5], gives a density guarantee of $\frac{2}{5}$. It is possible to directly use this algorithm to pack circles into a square in an online situation with a density of $\frac{\pi}{10} \approx 0.3142$. It would be interesting to see whether some form of online Split Packing would give better results.

A related problem asks for the smallest area so that we can always **cover the container** with circles of that combined area. For example, we conjecture that for an isosceles right triangle, any circle instance with a total area of at least its excircle's area is sufficient to cover it.

Acknowledgements

We thank the three anonymous reviewers for their helpful comments.

References

[1] Brian Brubach. "Improved Bound for Online Square-into-Square Packing". In: *Approximation and Online Algorithms*. Springer, 2014, pp. 47–58.

[2] Ignacio Castillo, Frank J. Kampas, and János D. Pintér. "Solving circle packing problems by global optimization: numerical results and industrial applications". In: *European Journal of Operational Research* 191(3) (2008), pp. 786–802.

[3] Erik D. Demaine, Sándor P. Fekete, and Robert J. Lang. "Circle Packing for Origami Design is Hard". In: *5th International Conference on Origami in Science, Mathematics and Education*. AK Peters/CRC Press. 2011, pp. 609–626.

[4] Sándor P. Fekete and Hella-Franziska Hoffmann. "Online Square-into-Square Packing". In: *APPROX-RANDOM*. 2013, pp. 126–141.

[5] Sándor P. Fekete and Hella-Franziska Hoffmann. "Online Square-into-Square Packing". In: *Algorithmica* 77(3) (2017), pp. 867–901.

[6] Sándor P. Fekete, Sebastian Morr, and Christian Scheffer. "Split Packing: Algorithms for Packing Circles with Optimal Worst-Case Density". In: *CoRR* abs/1705.00924 (2017). http://arxiv.org/abs/1705.00924.

[7] Mhand Hifi and Rym M'hallah. "A literature review on circle and sphere packing problems: models and methodologies". In: *Advances in Operations Research* Article ID 150624 (2009).

[8] Pedro Hokama, Flávio K. Miyazawa, and Rafael C. S. Schouery. "A bounded space algorithm for online circle packing". In: *Information Processing Letters* 116(5) (May 2016), pp. 337–342. ISSN: 0020-0190.

[9] Robert J. Lang. "A computational algorithm for origami design". In: *Proceedings of the Twelfth Annual Symposium on Computational Geometry (SoCG)* (1996), pp. 98–105.

[10] Marco Locatelli and Ulrich Raber. "Packing equal circles in a square: a deterministic global optimization approach". In: *Discrete Applied Mathematics* 122(1) (2002), pp. 139–166.

[11] Flávio K. Miyazawa, Lehilton L.C. Pedrosa, Rafael C.S. Schouery, Maxim Sviridenko, and Yoshiko Wakabayashi. "Polynomial-time approximation schemes for circle packing problems". In: *European Symposium on Algorithms (ESA)*. Springer. 2014, pp. 713–724.

[12] John W. Moon and Leo Moser. "Some packing and covering theorems". In: *Colloquium Mathematicae*. Vol. 17. 1. Institute of Mathematics, Polish Academy of Sciences. 1967, pp. 103–110.

[13] Sebastian 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)*. 2017, pp. 99–109.

[14] 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*. 1992, pp. 45–54.

[15] Eckard Specht. *Packomania*. 2015. URL: http://www.packomania.com/.

[16] Kokichi Sugihara, Masayoshi Sawai, Hiroaki Sano, Deok-Soo Kim, and Donguk Kim. "Disk packing for the estimation of the size of a wire bundle". In: *Japan Journal of Industrial and Applied Mathematics* 21(3) (2004), pp. 259–278.

[17] Péter Gábor Szabó, Mihaly Csaba Markót, Tibor Csendes, Eckard Specht, Leocadio G. Casado, and Inmaculada García. *New Approaches to Circle Packing in a Square*. Springer US, 2007.

[18] Huaiqing Wang, Wenqi Huang, Quan Zhang, and Dongming Xu. "An improved algorithm for the packing of unequal circles within a larger containing circle". In: *European Journal of Operational Research* 141(2) (Sept. 2002), pp. 440–453. ISSN: 0377-2217.

[19] D. Würtz, M. Monagan, and R. Peikert. "The history of packing circles in a square". In: *Maple Technical Newsletter* (1994), pp. 35–42.

[20] Yinfeng Xu. "On the minimum distance determined by $n(\leq 7)$ points in an isoscele right triangle". In: *Acta Mathematicae Applicatae Sinica* 12(2) (1996), pp. 169–175.