Approximating the Packing of Unit Disks into Simple Containers*

Helmut Alt and Nadja Seiferth

Institute for Computer Science, Freie Universität Berlin
lastname@mi.fu-berlin.de

Abstract

We study the following problems: Given a triangle or parallelogram, how many unit-disks can be packed without overlap into it? It is not known whether these problems are NP-hard or in NP. We give the first results on their approximability and therefore a better understanding of their complexity. In the case that all inner angles are bounded from below by a constant, we give a PTAS for each problem. For the case of arbitrarily small inner angles, we give an algorithm with constant-factor approximation and polynomial running time for triangles.

1 Introduction

Efficient disjoint packing of disks and spheres has been considered a natural and challenging problem in geometry for centuries. Kepler's famous conjecture about the most dense packing of spheres in three-dimensional space has been proven after nearly three hundred years by Hales and his group [5]. The densest packing of unit radius disks in the plane being the hexagonal grid has been known before. Earliest attempts for a proof go back to Lagrange although a rigorous proof was eventually given by Fejes-Tóth [4].

Algorithmic questions arise if instead of the whole space certain finite *containers* and sets of spheres or disks are considered. In the web, data bases can be found of the smallest containers (circles, squares etc.) which are known to hold n unit disks for n from 1 to several thousands [1].

We consider the converse problem of packing a maximal number of unit disks into a given finite size container. Even for the simplest container shapes such as disks or squares the problem is neither known to be solvable in polynomial time nor to be NP-hard. One major problem is that the number of disks to be computed can be exponential in the input size.

In a previous article, we developed PTAS for circular and square containers [2]. A natural representation for more general shapes would be simple or convex polygons. This paper is a small first step in this direction, showing a constant factor approximation algorithm for triangles. We can find PTAS for triangles and parallelograms if the containers are *fat*, i.e., their smallest angle is bounded from below by a constant.

2 First Order Theory of the Reals

We first observe that the decision problem, whether n unit circles can be packed into some container C is *decidable* if the shape of C can be described by finitely many polynomial inequalities. In fact, in this case the problem can be described by a formula in the first order theory of the reals. This formula contains $2n$ variables $s_1, t_1, ..., s_n, t_n$ which represent

* Supported by the German Science Foundation within the collaborative DACH project *Arrangements and Drawings* as DFG Project MU3501/3-1.

36th European Workshop on Computational Geometry, Würzburg, Germany, March 16–18, 2020.
This is an extended abstract of a presentation given at EuroCG'20. 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.

the coordinates where to place the centers of the unit circles. Moreover, the following conditions have to be stated in the formula:

1. all unit circles lie completely inside C
2. no two unit circles may intersect

Clearly, if C has a constant size description, condition 1 is the conjunction of $O(n)$ and condition 2 the conjunction of $O(n^2)$ formulas of constant size. The final formula is obtained by putting $\exists s_1, t_1, ..., s_n, t_n$ in front of these conditions.

The truth of first order formulas of the reals was shown to be decidable by Tarski in the 1950s. The fastest algorithm for this task is given by Basu et al. [3], Theorem 1.3.2. Plugging in the parameters of our problem, we obtain that the number of arithmetic operations to solve the problem is $n^{O(n)}$.

In order to determine the maximum number n_{max} of unit disks that can be packed into C we can make an exponential and binary search for n_{max} involving the decision algorithm in each of the $\log(n_{max})$ steps. If n_{max} is bounded by a constant we obtain:

▶ **Lemma 2.1.** *Suppose a container C described by constantly many polynomial inequalities is given by the coefficients of these polynomials, which are are rational numbers in binary notation. Then, if the maximum number n_{max} of unit circles that can be packed into C is bounded by a constant, it can be computed in polynomial time.*

It should be mentioned that the runtime of the algorithm is exponential in n_{max}, however.

3 Fat Containers

3.1 Parallelograms

We first consider the construction of a PTAS for determining the number of unit disks that can be packed into a *fat* parallelogram P. “Fat” means that the smaller angle γ of P is bounded from below by some constant γ_0. We will assume without loss of generality that two sides of P are parallel to the x-axis. Let a be the length of these sides. The input to the algorithm is a and (b_x, b_y) which are the coordinates of the left upper vertex of P, assuming that the left lower vertex of P is at the origin. These values are represented by rational numbers, where enumerator and denominator are given as binary integers.

Our **algorithm** has some similarity to the general technique by Hochbaum and Maass [6]. It subdivides P by lines parallel to its sides into congruent rhombi whose side length is some constant w to be determined later, see Figure 1. Since the rhombi will be truncated at the upper and right boundaries of P we get (up to) four different shapes of cells into which P is subdivided. For each of the shapes 1,..,4 we can determine the maximum number of unit disks it can hold in polynomial time using the algorithm given in Section 2.

Finally, we multiply the numbers obtained with the number of occurrences of the corresponding shapes (which can be computed by standard arithmetic), sum up, and return the obtained value as an approximation for the maximum number of unit disks that can be packed into P.

It remains to be shown that this algorithm is a PTAS for our problem if the input parallelogram P is fat. More precisely, for each $\varepsilon > 0$ it is possible to find a value for the parameter w such that the number n_{app} returned is at least $(1-\varepsilon)n_{max}$ where n_{max} is the maximum number of unit disks that can be packed into P.

To see this we consider an optimal solution OPT and assign its disks to the cells of the decomposition as follows: If a disk is contained in a cell, assign it to that cell. If it intersects only the horizontal boundary of cells, assign it to the lower cell that it intersects. If it

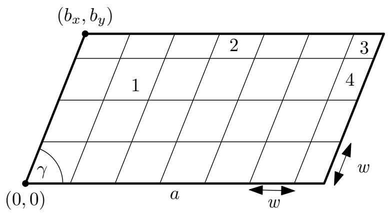

Figure 1 Parallelogram divided into cells defined by rhombus-grid with side length w.

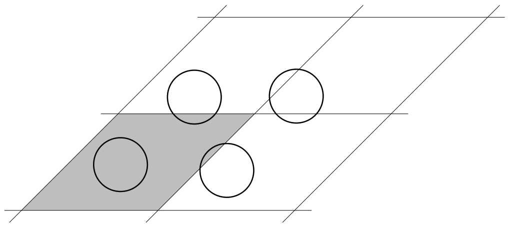

Figure 2 Disks assigned to a cell.

intersects only the vertical boundary of cells, assign it to the leftmost cell that it intersects. If it intersects a horizontal and a vertical grid-line, assign it to the cell at the lower left of the intersection of these two grid-lines, see Figure 2. In other words, all disks completely lying inside a cell q extended by a strip of width 2 to the top and to the right are assigned to q. For a cell q let n_i^q be the number of disks completely contained in q and n_b^q the disks assigned to it but not completely contained in it.

All n_i^q disks that are put completely inside a decomposition cell q by OPT are accounted for in n_{app} since there the maximum number n_m fitting into q is taken. However, the n_b^q disks assigned to but not completely contained in q are disregarded. Each of these disks must completely lie in a strip of width 4 around q. Since this strip has area $O(w)$ if the parallelogram P is fat we have $n_b^q = O(w)$.

On the other hand, Figure 3 shows that if $w \geq 2$ then $n_m \geq \lfloor (w \sin\gamma)/2 \rfloor^2 = \Omega(w^2 \sin^2\gamma)$ which is $\Omega(w^2)$ if P is fat. So we have $n_b^q / n_i^q \leq c/w$ for some constant c and for each cell q. Summing over all cells q we have

$$n_{max} = \sum_q (n_b^q + n_i^q) \leq (1 + c/w) \sum_q n_i^q \leq (1 + c/w) n_{app}$$

Choosing $w \geq d/\varepsilon$ for a suitable constant d yields:

▶ **Theorem 3.1.** *The algorithm described is a PTAS for the problem of finding the maximum number of unit disks that can be packed into a fat parallelogram P.*

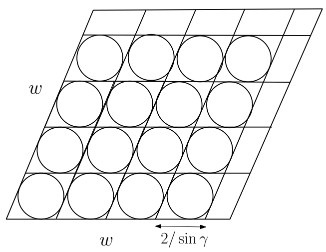

Figure 3 Lower Bound on the number of unit disks that can be packed into q

3.2 Triangles

Next, we will develop a PTAS for fat triangles, i.e., all of their angles are bounded from below by some constant α_0. Suppose that a triangle T is given by the lengths of its sides a, b, c with $a \geq b \geq c$. Side a is, without loss of generality, parallel to the x-axis and has its left vertex at the origin. Similarly to the case of parallelograms we decompose T into constant size cells by sets of lines which are parallel to one of two chosen sides a, b of T, see Figure 4. The distance of these lines is chosen such that both sides are divided into an equal

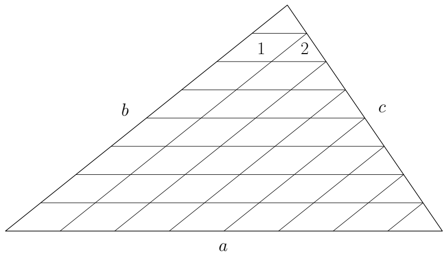

Figure 4 Dividing a triangle into cells with a parallelogram-grid with side lengths $\frac{a}{g}, \frac{b}{g}$ yields only two cell-types. Here $g = 8$.

number g of parts. As a consequence, there are only two types of congruent cells, which are parallelograms and triangles. As before, our algorithm computes the exact maximum number of disks fitting into each cell type by the technique of Section 2, multiplies it with the number of occurrences of the cell type and returns the sum of these numbers.

Analogousy to the corresponding proof for parallelograms, assigning disks of the optimal solution to the cells, it can be shown that for a given $\varepsilon > 0$ there is a choice of g, namely proportional to $b \cdot \varepsilon$ such that our algorithm computes a number n_{app} which is at least $(1 - \varepsilon)n_{max}$, i.e., we have:

▶ **Theorem 3.2.** *The algorithm described is a PTAS for the problem of finding the maximum number of unit disks that can be packed into a fat triangle T.*

4 Arbitrary Triangular Containers

For arbitrary triangular containers T, we can still construct in polynomial time a constant factor approximation for the maximum number n_{max} of unit disks that can be packed into T.

As before, let a be the longest side of T (we will also use a for the length of a), suppose again that it is parallel to the x-axis with its left vertex being the origin. Let h be the height of T perpendicular to a.

The idea of the algorithm is as follows: We first check whether $n_{max} = 0$ or $n_{max} = 1$. If not we imagine decomposing T into slabs of height 2 by lines parallel to a. Then the widths of the slabs are $s_1, ..., s_k$ where $s_i = a - 2i \cdot a/h$ for $i = 0, \cdots k$, see Figure 5.

Let k be the largest integer for which $s_k \geq 2$, namely $k = \lfloor (a-2)h/(2a) \rfloor$. In each slab the rectangle where the height of the slab is equal to 2 is filled contiguously with unit disks starting from the left end. Observe, that we cannot do this construction explicitely since the number of disks involved is exponential in the (bit-)size of the input.

However, the total number of disks packed can be computed, which is

$$\sum_{i=1}^{k} \lfloor s_i/2 \rfloor > \sum_{i=1}^{k} (s_i/2 - 1) > \lfloor ka/2 - ak(k+1)/(2h) - k \rfloor$$

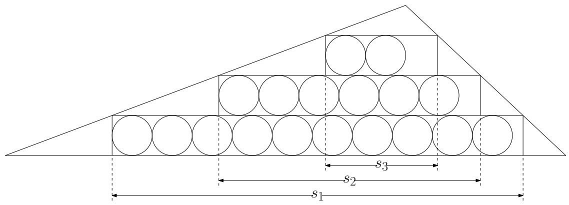

Figure 5 Disks packed in layers of height 2 into a general triangle.

The algorithm therefore returns $n_{app} = max(k, \lfloor ka/2 - ak(k+1)/(2h) - k \rfloor)$ as an approximation of n_{max}. As can be easily seen, this is at least half of the true number of disks packed.

It remains to be shown that n_{app} is only by a constant factor smaller than n_{max}.

Let us call a placement of a set of unit disks *maximal* (with respect to T) if no unit disk inside T can be added to the placement without intersecting others. Therefore, any disk of an optimal packing must be intersected (properly) by a disk of a maximal placement. On the other hand, since the kissing number of disks is 6, any disk of the maximal placement can intersect at most 5 disks of the optimal packing. Therefore, if there is a maximal placement of size n than the maximum packing has size at most $5n$.

As can be seen, the packing in Figure 5 is not yet maximal. We achieve this by an enlarged placement (which is not a packing any more) as shown in Figure 6: all layers

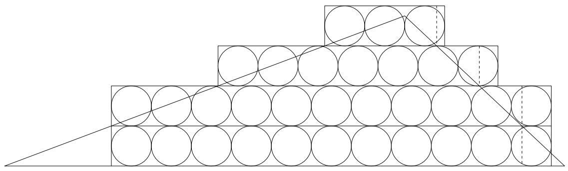

Figure 6 Disks placed in layers of height 2 onto a general triangle.

$i = 1, \cdots, k$ are moved up one level and an additional disk is added at their right end. Only the new layer 1 is the old one with the added disk. As can be easily seen, this placement is maximal and has at most three times as many disks as the previous packing.

Combining these ideas and summarizing we obtain:

▶ **Theorem 4.1.** *Given an arbitrary triangle T as input, the algorithm described returns a value n_{app} which is a constant factor approximation of the maximum number n_{max} of unit disks that can be packed into T.*

5 Conclusion

As was mentioned before, this contribution is only a small step towards determining complexity aspects of packing unit disks into given containers. Work in progress is the problem for arbitrary skinny parallelograms. In general, it would be interesting to gain more insight in the case of polygonal containers, even under the restriction that they are convex.

Acknowledgments. We thank Sándor Fekete for valuable hints concerning this reseach.

References

1 Packomania. URL: http://www.packomania.com/.

2 Helmut Alt and Nadja Scharf. Polynomial Time Approximation Schemes for Circle Packing Problems. In *Proceedings of the 33th European Workshop on Computational Geometry (EuroCG)*, pages 101–104, 2017. URL: http://csconferences.mah.se/eurocg2017/proceedings.pdf.

3 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the Combinatorial and Algebraic Complexity of Quantifier Elimination. *J. ACM*, 43(6):1002–1045, 1996.

4 L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum.* Springer Berlin Heidelberg, 2nd edition, 1972.

5 Thomas Hales, Mark Adams, Gertrud Bauer, Dat Dang, John Harrison, Truong Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Thang Nguyen, Truong Nguyen, Tobias Nipkow, Steven Obua, Joseph Pleso, Jason Rute, Alexey Solovyev, An Ta, Trân Trung, Diep Trieu, and Roland Zumkeller. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5, 2015.

6 Dorit S. Hochbaum and Wolfgang Maass. Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. *J. ACM*, 32(1):130–136, 1985.