3D-Disk-Packing*

Helmut Alt[1], Otfried Cheong[2], Ji-won Park[2], and Nadja Scharf[1]

1 **Institut für Informatik, Freie Universität Berlin**
`alt@mi.fu-berlin.de,nadja.scharf@fu-berlin.de`

2 **School of Computing, KAIST**
`otfried@kaist.edu,wldnjs1727@kaist.ac.kr`

Abstract

In this article, we consider the problem of finding in three dimensions a minimum volume axis-parallel cuboid container into which a given set of unit size disks can be packed under translations. The problem is neither known to be NP-hard nor to be in NP. We give a constant factor approximation algorithm based on reduction to finding a shortest Hamiltonian path in a weighted graph. As a byproduct, we can show that there is no finite size container into which all unit disks can be packed simultaneously.

1 Introduction

Packing a set of geometric objects in a nonoverlapping way into a minimum size container is an intriguing problem and because of its practical significance it has been widely investigated. For a survey see [1, 6] and the references therein. Even simple variants like packing a set of rectangles into a rectangular container turn out to be NP-hard [4]. Whereas that simple problem is in NP, in many cases not much is known about the true complexity of the problem.

Constant factor approximation algorithms of polynomial running time have been found for many variants of the packing, in particular for finding minimum size rectangular or convex containers for a set of convex polygons under translations [2], i.e., the objects may be translated but rotations are not allowed. Also, approximation algorithms for rigid motions (translations and rotations) are known in this case.

In three dimensions, approximation algorithms for packing cuboids or convex polyhedra into minimum volume cuboid or convex containers are known if rigid motions are allowed [3]. It remains an open problem whether this is possible for translations only. In this paper, we give a positive answer for a restricted set of possible objects, namely disks of unit radius and axis-parallel cuboid containers. So far, our approximation factor is forbiddingly high but it should be of theoretical interest that the problem, which is neither known to be NP-hard nor to be in NP, can be approximated in polynomial time at all.

Packing disks in 3D is meant in the following sense: We say that two disks *touch* if their intersection contains only one point and that two disks *intersect* if their intersection consists of more than one point. By *nonoverlapping*, we mean that no two disks intersect whereas it is allowed that two disks touch. The main problem we study in this work is then defined as follows:

▶ **Definition 1.1** (3D-3D-Disk-Packing)**.** Given a set of unit disks by their unit normal vectors in $\mathbb{R}^3$. The goal is to find

- an axis-parallel box of minimum volume such that all disks can be packed without overlapping under translation inside the box

* This work was partially supported by a fellowship within the FITweltweit program and by the Johann-Gottfried-Herder program of the German Academic Exchange Service (DAAD).

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. 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.

- and the actual packing of the disks inside the box.

We assume that no two disks are the same, i.e., no two normal vectors are parallel.

We will reduce approximating this problem with a constant factor to approximating the following problem with a constant factor.

▶ **Definition 1.2** (1D-3D-Disk-Packing). Given a set of nonidentical unit disks by their normal vectors in three dimensional space and an additional vector defining the direction of a line. The goal is to find an ordering of the disks with the following property: If the disks are placed nonoverlappingly with their centers in this order on the line, the distance from the center of the first to the center of the last disk is minimum. We call the distance of the center of the first to the center of the last disk when stabbed by the line the *length of the ordering*. See Figure 1 for a 2D example.

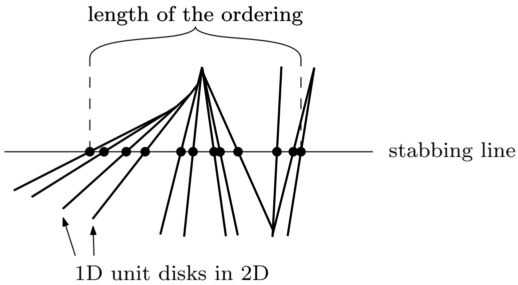

Figure 1 A (nonoptimal) solution to the 1D-2D-Interval-Packing problem. Here, the unit disks are unit line segments.

This problem then again will be reduced to finding the shortest Hamiltonian path in a complete weighted graph.

Let $a \in \mathbb{R}^3$ be a vector. Define $\mathrm{h}_a(D_1, D_2)$ to be the distance of the centers of the disks D_1 and D_2 when placed with their centers on a line parallel to the vector a such that D_1 and D_2 touch. $\mathrm{h}_a(D_1, D_2)$ can be computed easily from the normal vectors of D_1 and D_2 and it can be shown that $\mathrm{h}_a(D_1, D_2) = \mathrm{h}_a(D_2, D_1)$. The following lemma will be used for the reduction to Hamiltonian path.

▶ **Lemma 1.3.** *For disks D_1, D_2, D_3 and axis a, it holds that $\mathrm{h}_a(D_1, D_2) + \mathrm{h}_a(D_2, D_3) \geq \mathrm{h}_a(D_1, D_3)$, i.e., the triangle inequality holds.*

We omit the proof due to space constraints. It can be shown by contradiction, assuming that D_1 touches D_3 in a point x that is not part of D_2. Then considering the triangle formed by the centers of D_1 and D_3, and x, it can be shown that D_2 cannot fit between D_1 and D_3.

2 Approximation Algorithms

Next, we will show how to reduce the 1D-3D-Disk-Packing problem to finding the shortest Hamiltonian path in a complete weighted graph and obtain a constant factor approximation in this way. Afterwards we will use this approximation algorithm to compute a constant factor approximation for 3D-3D-Disk-Packing.

2.1 1D-3D-Disk-Packing Approximation

Algorithm 1 computes an approximate 1D-3D-Disk-Packing. In fact, since an ordering of the disks directly corresponds to a Hamiltonian path in G, the triangle inequality holds in G by

Lemma 1.3, and Hoogeveen's algorithm computes a $\frac{5}{3}$-approximation for it in polynomial time. So, we get the following theorem.

Input: n unit disks given by their normal vectors, vector a
Output: Ordering of the n disks

Generate complete weighted graph G with n vertices:
Set the weight of the edge (i, j) to $\mathrm{h}_a(D_i, D_j)$ for all $1 \leq i, j \leq n, i \neq j$;
For all $1 \leq i, j \leq n$ with $i \neq j$, approximate shortest Hamiltonian path on the graph with endpoints i and j with Hoogeveen's algorithm [5] and determine the overall shortest path;
return the ordering of the overall shortest path;

Algorithm 1: Approximation algorithm for 1D-3D-Disk-Packing

▶ **Theorem 2.1.** *Algorithm 1 computes a $\frac{5}{3}$-approximation for 1D-3D-Disk-Packing in polynomial time.*

In the next section, we will use Algorithm 1 to approximate 3D-3D-Disk-Packing.

2.2 3D-3D-Disk-Packing Approximation

We define $w_{\min}, d_{\min}, h_{\min}$ to be the maximum extension of any disk in x-,y-, and z-direction respectively and, thus, the minimum width, depth, and height any container for the disks must have. Let $w = s \cdot w_{\min}$ and $d = s \cdot d_{\min}$ for a constant $s > 1$ to be defined later. Algorithm 2 computes an approximate 3D-3D-Disk-Packing.

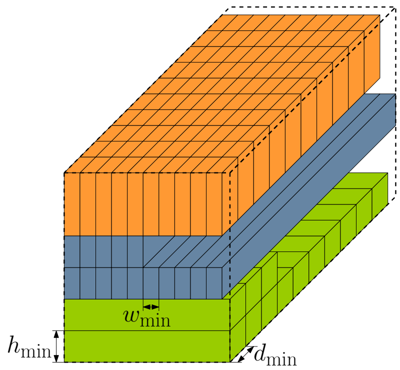

Figure 2 Example container for $s = 10.5$. The green boxes are the enlarged pieces obtained by dividing the container-box computed by Algorithm 1 for the disks in $\mathcal{X}$. Here, they form two layers. The blue boxes contain disks from $\mathcal{Y}$ and the orange boxes contain disks from $\mathcal{Z}$.

To analyze Algorithm 2 we first give a bound on W, D, and H. Observe that the angle between the normal vector of a disk and the axis it gets stabbed by in Algorithm 2 can be at most $\varphi = \arccos(\frac{1}{\sqrt{3}})$.

▶ **Lemma 2.2.** *It holds that*

$$W \leq 109 \cdot \frac{\mathrm{OPT}}{d_{\min} h_{\min}}, D \leq 109 \cdot \frac{\mathrm{OPT}}{w_{\min} h_{\min}}, H \leq 109 \cdot \frac{\mathrm{OPT}}{w_{\min} d_{\min}},$$

where OPT *is the volume of an optimal container.*

Input: n unit disks given by their normal vectors
Output: nonoverlapping packing of the disks into an axis-parallel box

1 Partition the n disks into three sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ according to the axis their normal vectors form the smallest angle with;
2 Call Algorithm 1 for the disks in $\mathcal{X}$ and vector $(1, 0, 0)$. If L_x is the length of the returned ordering, this can be interpreted as a packing of the disks in $\mathcal{X}$ into an axis-parallel box of width $W = L_x + w_{\min}$, depth $d_{\min}$, and height $h_{\min}$;
3 Analogously to Step 2 get packings for the disks in $\mathcal{Y}$ and $\mathcal{Z}$ into boxes of dimensions $w_{\min} \times D \times h_{\min}$ and $w_{\min} \times d_{\min} \times H$ respectively;
4 Divide the box obtained for $\mathcal{X}$ into pieces of width $w - w_{\min}$;
5 Assign each disk to the piece its point with smallest x-coordinate lies in;
6 Enlarge each piece from width $w - w_{\min}$ to width w such that all disks that are assigned to a piece are completely contained in that piece;
7 Divide the box obtained for $\mathcal{Y}$ into pieces of depth d analogously to Steps 4 to 6;
8 Divide the box obtained for $\mathcal{Z}$ into $\left\lfloor \frac{w}{w_{\min}} \right\rfloor \left\lfloor \frac{d}{d_{\min}} \right\rfloor$ pieces of width $w_{\min}$ and depth $d_{\min}$;
9 Analogously to Steps 5 and 6, enlarge the height of each piece by $h_{\min}$;
10 Arrange the pieces to a box of width w and depth d. The pieces containing disks of $\mathcal{X}$ form $\left\lceil \left\lceil \frac{W}{w - w_{\min}} \right\rceil / \left\lfloor \frac{d}{d_{\min}} \right\rfloor \right\rceil$ layers of height $h_{\min}$, the pieces containing disks of $\mathcal{Y}$ form $\left\lceil \left\lceil \frac{D}{d - d_{\min}} \right\rceil / \left\lfloor \frac{w}{w_{\min}} \right\rfloor \right\rceil$ layers of height $h_{\min}$, and the pieces containing disks from $\mathcal{Z}$ form one layer of height $H / \left(\left\lfloor \frac{w}{w_{\min}} \right\rfloor \left\lfloor \frac{d}{d_{\min}} \right\rfloor \right) + h_{\min}$ (See Figure 2 for an example);
11 **return** the resulting box with the packed disks;

Algorithm 2: Approximation algorithm for 3D-3D-Disk-Packing

Proof. Consider an optimal container with width W_{OPT}, depth D_{OPT}, and height H_{OPT} and let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be the partition of disks into subsets as in Algorithm 2. Furthermore consider a grid with some side length g on the x-z-plane and lines parallel to the y-axis through the grid cell centers. Then, each point has distance at most $\frac{g}{\sqrt{2}}$ to the closest line. So, every disk in $\mathcal{Y}$ is stabbed by a line in a point of distance at most $\frac{g}{\sqrt{2}\sin\left(\frac{\pi}{2} - \varphi\right)}$ from the disk center if g is small enough, i.e. $cg < 1$, where $c = \frac{1}{\sqrt{2}\sin\left(\frac{\pi}{2} - \varphi\right)} = \sqrt{\frac{3}{2}}$. See Figure 3 for illustration. Therefore, each disk in $\mathcal{Y}$ contains a disk of radius $1 - cg$ stabbed by a line through its center.

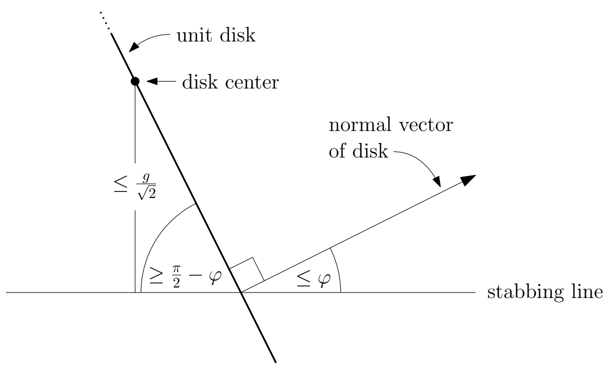

Figure 3 Distance of a disk center to the stabbing line

So, by placing the line segments that are the intersection of the container and the lines behind each other so that they touch, we get a solution to the 1D-3D-Disk-Packing-Problem for the disks in $\mathcal{Y}$ but with radius $1 - cg$. By stretching this solution by $1/(1-cg)$, we get a solution for disks of radius 1. Let $L_{\mathrm{OPT}_\mathcal{Y}}$ be the length of an optimal solution for the 1D-3D-Disk-Packing problem for the disks in $\mathcal{Y}$. Then, this length can be at most the length of our solution, i.e.,

$$L_{\mathrm{OPT}_\mathcal{Y}} \leq \left\lceil \frac{H_{\mathrm{OPT}}}{g} \right\rceil \left\lceil \frac{W_{\mathrm{OPT}}}{g} \right\rceil D_{\mathrm{OPT}} \cdot \frac{1}{1-cg}.$$

By using $w_{\min}, h_{\min} \leq 2$ and $W_{\mathrm{OPT}} \geq w_{\min}, H_{\mathrm{OPT}} \geq h_{\min}$, it can be shown that

$$L_{\mathrm{OPT}_\mathcal{Y}} \leq \frac{(g+2)^2}{g^2(1-cg)} \cdot \frac{\mathrm{OPT}}{w_{\min} h_{\min}}. \tag{1}$$

Since we use Algorithm 1 to compute a 1D-3D-Disk-Packing solution for $\mathcal{Y}$, we get by Theorem 2.1

$$D \leq \frac{5}{3} \cdot L_{\mathrm{OPT}_\mathcal{Y}} + d_{\min},$$

where the extra term $d_{\min}$ comes from the fact that the length of a 1D-3D-Disk-Packing is defined as the distance of the center of the first disk to the center of the last disk and we are interested in the total depth of the packing. By inequality (1),

$$D w_{\min} h_{\min} \leq \left(\frac{5(g+2)^2}{3g^2(1-cg)} + 1 \right) \mathrm{OPT}.$$

Optimizing for g yields $g = \sqrt{\frac{1}{3}\left(27 + 4\sqrt{6}\right)} - 3$ and a factor of approximately 108.49. The calculations for W and H are analogous. This implies the lemma. ◀

Now, we are ready to state the main theorem of this article.

▶ **Theorem 2.3.** *Algorithm 2 computes a 593-approximation for 3D-3D-Disk-Packing in polynomial time.*

Proof. The container computed by Algorithm 2 is a box with base area $w \cdot d$ and height $\left\lceil \left\lceil \frac{W}{w - w_{\min}} \right\rceil / \left\lfloor \frac{d}{d_{\min}} \right\rfloor \right\rceil h_{\min} + \left\lceil \left\lceil \frac{D}{d - d_{\min}} \right\rceil / \left\lfloor \frac{w}{w_{\min}} \right\rfloor \right\rceil h_{\min} + H / \left(\left\lfloor \frac{w}{w_{\min}} \right\rfloor \left\lfloor \frac{d}{d_{\min}} \right\rfloor \right) + h_{\min}$ (See step 10 in Algorithm 2). Using Lemma 2.2, the definition of w and d (see the beginning of this section), and $w_{\min} d_{\min} h_{\min} \leq \mathrm{OPT}$ it can be shown that the volume of the container is at most

$$s^2 \left(\frac{2 \cdot \frac{109}{s-1} + 1}{s-1} + \frac{109}{(s-1)^2} + 3 \right) \mathrm{OPT}.$$

Optimizing for s gives a long term as approximation factor that is smaller than 593. ◀

3 Unbounded containers are necessary

In this section, we will conclude from our previous results that there is no bounded size container into which all unit disks can be packed. More precisely, we will show:

▶ **Theorem 3.1.** *Packing a set of n unit disks requires a container of size $\Omega(\sqrt{n})$ in the worst case.*

Proof. In the following, we will show that $\Omega(\sqrt{n})$ is a lower bound for the container constructed by Algorithm 2 which is within a constant factor of the optimal container. From that the theorem follows immediately.

Identify any unit disk with its normal vector in the unit sphere S^2. Consider a sufficiently small rectangular surface patch $P = I_1 \times I_2 \subset S^2$ where I_1, I_2 are nonempty intervals of spherical coordinates. Let P be symmetric to the equator and I_1 and I_2 sufficiently small, so that all disks corresponding to points in P are stabbed by the same axis in Algorithm 2. Furthermore, for any two points in P the shorter grand circle segment connecting them should lie completely inside P. For a given $\varepsilon > 0$, subdivide P by horizontal and vertical lines at distance ε, yielding a grid of points in P of size $n \geq c_1/\varepsilon^2$ for some constant $c_1 > 0$. Let A be the set of unit disks corresponding to the grid points. With standard geometric arguments it is possible to prove the following

▶ Claim 3.2. There is a constant $c_2 > 0$ such that for any two grid points having distance δ on S^2 the centers of the corresponding unit disks have distance at least $c_2\delta$ when stabbed consecutively on a line as in Algorithm 1.

Now observe, that if P is chosen close enough to the equator, any two distinct points in A have distance at least $c_3\varepsilon$ for some constant $c_3 > 0$. Therefore, by the previous claim the distance of the centers of the corresponding unit disks, when stabbed consecutively, is at least $c_4\varepsilon$ for some constant $c_4 > 0$. Consequently the length of a line segment stabbing all disks in A must be at least $c_4\varepsilon(n-1)$. Since $n \geq c_1/\varepsilon^2$, this is in $\Omega(\sqrt{n})$ as ε tends to 0. From Lemma 2.2 follows that this is also a lower bound for the volume of a container computed by Algorithm 2. ◀

From Theorem 3.1, we obtain immediately

▶ **Corollary 3.3.** *There is no finite size container into which all unit disks can be packed.*

This seems obvious at first glance, but observe that for the case of one-dimensional objects it is false. In fact, all unit length line segments can be packed in arbitrary dimension $d \geq 2$ into a container of finite size for example by placing them with one endpoint in the origin.

References

1. Helmut Alt. Computational aspects of packing problems. *Bulletin of the EATCS*, 118, 2016.
2. Helmut Alt, Mark de Berg, and Christian Knauer. Approximating Minimum-Area Rectangular and Convex Containers for Packing Convex Polygons. *JoCG*, 8(1):1–10, 2017.
3. Helmut Alt and Nadja Scharf. Approximating Smallest Containers for Packing Three-Dimensional Convex Objects. In *Proc. 27th Annu. Internat. Sympos. Algorithms Comput. (ISAAC)*, 2016.
4. Robert J. Fowler, Mike Paterson, and Steven L. Tanimoto. Optimal Packing and Covering in the Plane are NP-Complete. *Inf. Process. Lett.*, 12(3):133–137, 1981.
5. J. A. Hoogeveen. Analysis of Christofides' Heuristic: Some Paths Are More Difficult Than Cycles. *Oper. Res. Lett.*, 10(5):291–295, 1991.
6. Guntram Scheithauer. *Zuschnitt- und Packungsoptimierung: Problemstellungen, Modellierungstechniken, Lösungsmethoden.* Vieweg+Teubner Verlag, 2008.