Online Circle and Sphere Packing*

Carla Negri Lintzmayer[1], **Flávio Keidi Miyazawa**[2], **Eduardo Candido Xavier**[2]

[1]Center for Mathematics, Computation and Cognition – Federal University of ABC
Av. dos Estados 5001, Santo André, SP, Brazil

[2]Institute of Computing – University of Campinas
Av. Albert Einstein 1251, Campinas, SP, Brazil

carla.negri@ufabc.edu.br, {fkm,eduardo}@ic.unicamp.br

Abstract. *In the Online Circle Packing in Squares, circles arrive one at a time and we need to pack them into the minimum number of unit square bins. We improve the previous best-known competitive ratio for the bounded space version from* 2.439 *to* 2.3536 *and we also give an unbounded space algorithm. Our algorithms also apply to the Online Circle Packing in Isosceles Right Triangles and Online Sphere Packing in Cubes, for which no previous results were known.*

1. Introduction

We consider the Online Circle Packing in Squares, Online Circle Packing in Isosceles Right Triangles, and Online Sphere Packing in Cubes problems. They receive an online sequence of circles/spheres of different radii and the goal is to pack them into the minimum number of unit squares, isosceles right triangles of leg length one, and unit cubes, respectively. By packing we mean that two circles/spheres do not overlap and each one is totally contained in a bin. As applications we can mention origami design, crystallography, error-correcting codes, coverage of a geographical area with cell transmitters, storage of cylindrical barrels, and packaging bottles or cans [Szabó et al. 2007].

In online packing problems, one item arrives at a time and must be promptly packed, without knowledge of further items. Also, after packing an item, it cannot be moved to another bin. At any moment a bin is open or closed: we can only pack items into open bins and, once a bin is closed, it cannot be opened again. The algorithms can have *bounded space*, when the number of open bins is bounded by a constant, or *unbounded space*, when there is no guarantee on this number. Usually, items cannot be reorganized, but our unbounded space algorithms may do this on a constant number of items.

Several works consider items to be squares or rectangles [Christensen et al. 2017], but the Online Circle Packing in Squares was considered only in [Hokama et al. 2016], who showed a lower bound of 2.292 on the competitive ratio of any bounded space algorithm and gave one with asymptotic competitive ratio 2.439. Offline Circle Packing in Squares was considered in [Miyazawa et al. 2016] and there are several results for packing circles or spheres of same radii [Szabó et al. 2007, Tatarevic 2015].

We show bounded and unbounded space algorithms which work for our three problems. The bounded space ones are based on the one given in [Hokama et al. 2016],

*This work was supported by São Paulo Research Foundation (grants 2016/14132-4, 2015/11937-9, 2016/01860-1, 2016/23552-7) and National Counsel of Technological and Scientific Development (grants 306358/2014-0, 311499/2014-7, and 425340/2016-3).

but they have a simpler analysis, we improved the occupation ratio for the class of small circles, we subdivide the bins in a simpler way, and we used another method to find the numerical results. The unbounded space algorithms are a modification of the bounded space ones using an idea presented in [Epstein 2010]. We also give lower bounds on the competitive ratio of any bounded space algorithm for Online Circle Packing in Isosceles Right Triangles and Online Sphere Packing in Cubes. Due to space constraints, in the next section we only describe our algorithms for Online Circle Packing in Squares. The same algorithms, with relatively minor changes work for the other two problems[1].

2. Online Circle Packing in Squares

For an integer M given as a parameter, a circle of radius r is *large* if $r > 2/M$ and it is *small* if $r \leq 2/M$. Large and small circles are packed separately, into *Lbins* and *Sbins*, respectively, which are, nonetheless, unit squares. The only difference between the bounded and unbounded space algorithm is how they pack some of the large circles.

For an integer $i \geq 1$, let ρ_i* be the largest value such that i circles of radius ρ_i* can be packed into a unit square. Let ρ_i be ρ_i*, if $i \leq 30$, or the best-known lower bound for ρ_i*, if $30 < i \leq 9996$ [Specht]. Let $K \in \mathbb{Z}$ be such that $\rho_{K+1} < 2/M \leq \rho_K$. A large circle of radius r has type i, for $1 \leq i < K$, if $\rho_{i+1} < r \leq \rho_i$, or it has type K, if $2/M < r \leq \rho_K$. We denote a large circle of type i as I_i. Let $C \in \mathbb{Z}$ be a parameter. Given a small circle of radius r, we find the largest integer p such that $C^p r \leq 2/M$. We then classify such circle as type (i, p) if $2/(i+1) < C^p r \leq 2/i$, where $M \leq i < CM$.

The hexagonal packing is the densest one for circles of equal radii and we will use it to pack small circles. For each i, with $M \leq i < CM$, an Sbin is of type i if it is used to pack small circles of type (i, p), for any $p \geq 0$. We keep at most one Sbin of type i open at a time. A *q-bin*(i, p) is a square of side length $1/C^{p+1}$. When an Sbin of type i is opened, it is divided into C^2 q-bins$(i, 0)$. A q-bin(i, p) is either subdivided into C^2 q-bins$(i, p+1)$ or it is tiled with hexagons of side $4/(C^p i\sqrt{3})$ to pack small items of type (i, p) (because a circle of radius at most r fits into a hexagon of side length $2r/\sqrt{3}$). Note that we must have $1/C^{p+1} \geq 2/(iC^p)$ for this to work, but since $i < CM$, it suffices to choose $M > 2$.

When a small item of type (i, p) arrives, the algorithms will simply try to pack it into a hexagon of a q-bin(i, p). If there is no such hexagon, they will find an empty q-bin(i, p') (not tiled yet) with the largest p' such that $p' < p$ and subdivide it, until a q-bin(i, p) is found (at which point it will be tiled). Next theorem shows the occupation ratio of a closed Sbin, which is the minimum total area occupied by the circles packed in such bin. Parameter C is chosen to be 5 because it maximizes this ratio.

Theorem 1 *The occupation ratio of a closed Sbin of type* i*, for* $M \leq i < 5M$*, is at least* $\frac{551}{600}\left(1 - \frac{43.1}{M} + \frac{18.48}{M^2}\right)\frac{\pi}{\sqrt{12}}\frac{M^2}{(M+1)^2}$.

Now for packing large circles, since $\rho_{9996} < 0.005076143$ and $\rho_i < \rho_{9996}$ for $i < 9996$, we chose $M = 360$, which makes $2/M \leq 0.005555556$. Thus, we can find K such that $\rho_{K+1} < 2/M \leq \rho_K$, and so we can classify all large items. For each i, with $1 \leq i \leq K$, an Lbin is of type i if it packs large circles of type i. At most one Lbin of type i is kept open at a time. When an Lbin of type i is opened, it is divided into i *c-bins*, which are circles of radii ρ_i. When an I_i arrives, it is either packed into an empty c-bin

[1] A full version is available at `https://arxiv.org/abs/1708.08906`.

or the current Lbin of type i (if any) is closed and a new one is opened. Note that at most K Lbins are kept open by the algorithm at any given time. Together with the $(C-1)M$ Sbins, we can see that this algorithm has bounded space. Furthermore, note that each closed Lbin of type i has occupation ratio of at least $i\pi\rho_{i+1}^2$.

The analysis of the competitive ratio uses the weighting method [Epstein 2010], in which we show a weighting function w over the items such that the sum of weights of items in any bin is at least 1, except for a constant number of bins. Afterwards, we find every feasible configuration, i.e., sets of items that can be packed into a bin, calculate their sum of weights, and find the supremum β of such sums. As a result, the asymptotic competitive ratio of the algorithm is bounded from above by β. However, it is not reasonable to list every possible feasible configuration, so we use the following fact. Consider a bin of maximum weight with some known-circles $\mathcal{I}' \subseteq \mathcal{I}$ whose sum of weights is W and sum of areas is A, and let OR be a lower bound on the occupation ratio of the unknown circles of the bin (in $\mathcal{I}\backslash\mathcal{I}'$). If the weighting function is such that the weight of a circle $i \notin \mathcal{I}'$ of area $a(i)$ is at most $a(i)/\mathrm{OR}$, then the total sum of weights in such bin is at most $W + (1-A)/\mathrm{OR}$. With this, we can find only a few configurations and the lower bound on the occupation ratio of circles that are not on them, in order to find an upper bound on the asymptotic competitive ratio of our algorithms. For that, we use a two-phase program. The first phase uses constraint programming to give a set $\mathcal{F}$ of feasible configurations that contain only large circles. Each configuration $\mathcal{I} \in \mathcal{F}$ is associated with an integer $f(\mathcal{I})$ which indicates that all large circles from type 1 to $f(\mathcal{I})$ were tested to be part of $\mathcal{I}$. In the second phase the idea is to add circles of type greater than $f(\mathcal{I})$ in the remaining space of the bin by using a criterion of space: if the circle's area is at most the remaining area of the bin, such circle will be considered. For that we use an integer programming which simulates a knapsack problem. We may create an infeasible configuration, but this is not an obstacle since our goal is to find a configuration with maximum total weight. This two-phase program is used in the proofs of Theorems 2 and 3.

At last, we define $w(I_i) = \frac{1}{i}$, so a closed Lbin of type i has total weight $\sum_{j=1}^{i} \frac{1}{i} = 1$. Let $\mathrm{OR} = \frac{551}{600}\left(1 - \frac{43.1}{M} + \frac{18.48}{M^1}\right)\frac{\pi}{\sqrt{12}}\frac{M^2}{(M+1)^2}$. For a small circle c of type (i,p) and area $a(c)$, we define $w(c) = \frac{a(c)}{\mathrm{OR}}$, so a closed Sbin B of type i has total weight $\sum_{c\in B}\frac{a(c)}{\mathrm{OR}} = \frac{1}{\mathrm{OR}}a(B) \geq 1$, where $a(B)$ is the sum of areas of items in B and, according to Theorem 1, $a(B) \geq \mathrm{OR}$. Theorem 2 concludes with the asymptotic competitive ratio of the algorithm.

Theorem 2 *The algorithm for packing circles in squares with bounded space has asymptotic competitive ratio strictly below* 2.3536.

Packing large circles in the unbounded space algorithm follows an idea of [Epstein 2010]. A *waiting bin* packs either one I_1 with one I_2 (the I_1 at the left bottom corner and the I_2 at the right top corner) or one I_1 with two I_4 (the I_1 centered to the left border with one I_4 at the right bottom corner and the other at the right top corner), if their radius are related with a value D. Other circles are packed as before. The reorganization of packed items is allowed only for I_1s that are inside open waiting bins. The algorithm has unbounded space because we cannot guarantee how many waiting bins are open.

If an I_1 of radius $r > D$ arrives, then we pack it as in the bounded space algorithm (one per bin). Otherwise, $r \leq D$ and we pack it in an already opened waiting bin containing either one I_2 or at least one I_4, if one exists, or we open a new one to pack the I_1 and let it open waiting for an I_2 or two I_4. This last case is why we allow the reorganization of circles I_1 inside waiting bins. Let $\gamma = \sqrt{2}/(\sqrt{2}+1) - D$. If an I_2 of radius $r > \gamma$ arrives, then we pack it as in the bounded space algorithm (two per bin). When $r \leq \gamma$, the circles are labeled according to their arrival so that the following steps can be repeated at every 72 of them. If the I_2 is among the first 70, then it is packed as in the bounded space algorithm (two per bin); if it is one of the last 2, then it is packed in a waiting bin. Let $\lambda = 3/2 - \sqrt{2D+1}$. If an I_4 of radius $r > \lambda$ arrives, then we pack it as in the bounded space algorithm (four per bin). When $r \leq \lambda$, the circles are labeled according to their arrival so that the following steps are repeated at every 34 of them. If the I_4 is among the first 32, then we pack four per bin; if it is one of the last 2, then it is packed in a waiting bin. Values of γ and λ were chosen through simple algebraic expressions written considering the desired configurations of waiting bins. Since we need $\rho_2 < D \leq \rho_1$, $\rho_3 < \gamma \leq \rho_2$, and $\rho_5 < \lambda \leq \rho_4$ for them to be possible, and this is true for $\rho_2 < D < 0.331553$, we fixed $D = 0.325309$.

To analyse the competitive ratio of this algorithm we use a generalized weighting method [Epstein 2010]. We need to show weighting functions w_1 and w_2 whose sum of weights of items in any bin is at least 1 on average for at least one of the functions, except for a constant number of bins. The supremum β of the sums of weights of feasible configurations, for both functions, is an upper bound on the asymptotic competitive ratio.

When the algorithm ends, either there are open waiting bins with I_1, in which case we apply w_1 over all circles, or there are not, in which case we apply w_2. They differ from w only regarding I_1, I_2, and I_4 if their radii are at most D, γ, and λ, respectively. If I_1 has radius $r \leq D$, then $w_1(I_1) = 1$ and $w_2(I_1) = 0$. If I_2 has radius $r \leq \gamma$, then $w_1(I_2) = \frac{35}{72}$ and $w_2(I_2) = \frac{37}{72}$. If I_4 has radius $r \leq \lambda$, then $w_1(I_4) = \frac{8}{34}$ and $w_2(I_4) = \frac{9}{34}$. For both functions, the sum of weights in any bin is at least 1 if we do not consider I_1, I_2, and I_4, because this is true for w. Now consider w_1 was applied over such items. All waiting bins have total weight at least 1 because they contain an I_1. For every set of 72 I_2, we have 70 of them packed into 35 bins and the last 2 packed into 2 waiting bins with one I_1 each. Thus, the average weight of these bins is $\left(35\left(2\frac{35}{72}\right) + 2\left(1+\frac{35}{72}\right)\right)/37 = 1$. This is similar for I_4 and w_2. Theorem 3 concludes our result.

Theorem 3 *The algorithm for packing circles in squares with unbounded space has asymptotic competitive ratio strictly below* 2.3105.

References

Christensen, H. I., Khan, A., Pokutta, S., and Tetali, P. (2017). Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79.

Epstein, L. (2010). Two-dimensional Online Bin Packing with Rotation. *Theoretical Computer Science*, 411(31):2899–2911.

Hokama, P., Miyazawa, F. K., and Schouery, R. C. S. (2016). A Bounded Space Algorithm for Online Circle Packing. *Information Processing Letters*, 116(5):337–342.

Miyazawa, F. K., Pedrosa, L. L. C., Schouery, R. C. S., Sviridenko, M., and Wakabayashi, Y. (2016). Polynomial-Time Approximation Schemes for Circle and Other Packing Problems. *Algorithmica*, 76(2):536–568.

Specht, E. Packomania. `http://www.packomania.com/`. Accessed: 2018-03-27.

Szabó, P. G., Markót, M. C., Csendes, T., Specht, E., Casado, L. G., and García, I. (2007). *New Approaches to Circle Packing in a Square*. Springer Optimization and Its Applications. Springer US, New York, USA.

Tatarevic, M. (2015). On Limits of Dense Packing of Equal Spheres in a Cube. *The Electronic Journal of Combinatorics*, 22(1):35.