Approximation of Geometric Dispersion Problems *

(Extended Abstract)

Christoph Baur and Sándor P. Fekete

Center for Parallel Computing, Universität zu Köln, D–50923 Köln, Germany
[baur,fekete]@zpr.uni-koeln.de

Abstract. We consider problems of distributing a number of points within a connected polygonal domain P, such that the points are "far away" from each other. Problems of this type have been considered before for the case where the possible locations form a discrete set. Dispersion problems are closely related to packing problems. While Hochbaum and Maass (1985) have given a polynomial time approximation scheme for packing, we show that geometric dispersion problems cannot be approximated arbitrarily well in polynomial time, unless P=NP. We give a $\frac{2}{3}$ approximation algorithm for one version of the geometric dispersion problem. This algorithm is strongly polynomial in the size of the input, i. e., its running time does not depend on the area of P. We also discuss extensions and open problems.

1 Introduction: Geometric Packing Problems

In the following, we give an overview over geometric packing. Problems of this type are closely related to geometric dispersion problems, which are described in Section 2.

Two-dimensional packing problems arise in many industrial applications. As two–dimensional cutting stock problems, they occur whenever steel, glass, wood, or textile materials are cut. There are also many other problems that can be modeled as packing problems, like the optimal layout of chips in VLSI, machine scheduling, or optimizing the layout of advertisements in newspapers.

When considering the problem of finding the best way to pack a set of objects into a given domain, there are several objectives that can be pursued: We can try to maximize the value of a subset of the objects that can be packed and consider *knapsack problems*; we can try to minimize the number of containers that are used and deal with *bin packing problems*, or try to minimize the area that is used – in *strip packing problems*, this is done for the scenario where the domain is a strip with fixed width and variable length that is to be kept small.

All of these problems are NP-hard in the strong sense, since they contain the one-dimensional bin packing problem as a special case. However, there are

* This work was supported by the German Federal Ministry of Education, Science, Research and Technology (BMBF, Förderkennzeichen 01 IR 411 C7).

some important differences between the one- and the two-dimensional instances; and while there are many algorithms for one-dimensional packing problems that work well in practice (currently, benchmark instances of the one-dimensional knapsack problem with up to 250,000 objects can be solved optimally, see [31]), until recently, the largest solved benchmark instance of the two-dimensional orthogonal knapsack problem (i. e., packing rectangles into a rectangular container) had no more than 23 objects (see [3, 22]). One of the difficulties in two dimensions arises from the fact that an appropriate way of modeling packings is not easy to find; this is highlighted by the fact that the feasible space cannot be assumed to be convex. (Even if the original domain is convex, the remaining feasible space will usually lose this property after a single object is placed in the domain.) This makes it impossible to use standard methods of combinatorial optimization without additional insights. For an overview over heuristic and exact packing methods, see [40]. See [11–14, 38] for a recent approach that uses a combination of geometric and graph-theoretic properties for characterizing packings of rectangles and constructing relatively fast exact algorithms. Kenyon and Remila [24] give an "asymptotic" polynomial time approximation scheme for the strip packing problem, using the additional assumption that the size of the packed objects is insignificant in comparison to the total strip length. (In this context, see also [1].)

There are several other sources of difficulties of packing in two dimensions: the shape of the objects may be complicated (see [26] for an example from the clothing industry), or the domain of packing may be complicated. In this paper, we will deal with problems related to packing objects of simple shape (i. e., identical squares) into a *polygonal domain*: a connected region, possibly with holes, that has a boundary consisting of a total of n line segments, and the same number of vertices.

It should be noted that even when the structure of domain *and* objects are not complicated, only little is known – see the papers by Graham, Lubachevsky, and others [16, 19–21, 27–29] for packing identical disks into a strip, a square, a circle, or an equilateral triangle. Also, see Nelißen [34] for an overview of the so-called *pallet loading problem*, where we have to pack identical rectangles into a larger rectangle; it is still unclear whether this problem belongs to the class NP, since there may not be an optimal solution that can be described in polynomial time.

The following decision problem was shown to be NP-complete by Fowler et al. [15]; here and throughout the paper an *L-square* is a rectangle of size $L \times L$, and the number of vertices of a polygonal domain includes the vertices of all the holes it may have.

PACK(k, L):

Input: a polygonal domain P with n vertices, a parameter k, a parameter L.

Question: Can k many L-squares be packed into P?

PACK(k, L) is the decision problem for the following optimization problem:

$\max_k$PACK(L):
Input: a polygonal domain P with n vertices
Task: Pack k many L-squares into P, such that k is as big as possible.

It was shown by Hochbaum and Maass [23] that $\max_k$PACK(L) allows a polynomial time approximation scheme. The main contents of this paper is to examine several versions of the closely related problem

$\max_L$PACK(k):
Input: a polygonal domain P with n vertices
Task: Pack k many $L \times L$ squares into P, such that L is as big as possible.

2 Preliminaries: Dispersion Problems

The problem $\max_L$PACK(k) is a particular geometric *dispersion problem.* Problems of this type arise whenever the goal is to determine a set of positions, such that the objects are "far away" from each other. Examples for practical motivations are the location of oil storage tanks, ammunition dumps, nuclear power plants, hazardous waste sites – see the paper by Rosenkrantz, Tayi, and Ravi [36], who give a good overview, including the papers [6, 7, 9, 10, 18, 30, 32, 35, 39]. All these papers consider discrete sets of possible locations, so the problem can be considered as a generalized independent set problem in a graph. Special cases have been considered – see [5, 6]. However, for these discrete versions, the stated geometric difficulties do not come into play. In the following, we consider geometric versions, where the set of possible locations is given by a polygonal domain. We show the close connection to the packing problem and the polynomial approximation scheme by Hochbaum and Maass [23], but also a crucial difference: in general, if P≠NP, it cannot be expected that the geometric dispersion problem can be approximated arbitrarily well.

When placing objects into a polygonal domain, we consider the following problem, where $d(v, w)$ is the geodesic distance between v and w:

$$\max_{S \subset P, |S|=k} \min_{v,w \in S} d(v, w).$$

This version corresponds to the dispersion problems in the discrete case and will be called *pure dispersion.*

In a geometric setting, we may not only have to deal with distances between locations; the distance of the dispersed locations to the boundary of the domain can also come into play. This yields the problem

$$\max_{S \subset P, |S|=k} \min_{v,w \in S} \{d(v, w), d(v, \partial P)\},$$

where ∂P denotes the boundary of the domain P. This version will be called *dispersion with boundaries.*

Finally, we may consider a generalization of the problem $\max_L$PACK(k), which looks like a mixture of both previous variants:

$$\max_{S \subset P, |S|=k} \min_{v,w \in S} \{2d(v,w), d(v, \partial P)\}.$$

Since this corresponds to packing k many d-balls of maximum size into P, this variant is called *dispersional packing.*

It is also possible to consider other objective functions. Maximizing the average distance instead of the minimum distance can be shown to lead to a one-dimensional problem for pure dispersion (all points have to lie on the boundary of the convex hull of P). Details are omitted from this abstract.

Several distance functions can be considered for $d(v,w)$; the most natural ones are L_2 distances and L_1 or L_∞ distances. In the following, we concentrate on rectilinear, i.e., L_∞ distances. This means that we will consider packing squares with edges parallel to the coordinate axes. Most of the ideas carry over for L_2 distances by combining our ideas with the techniques by Hochbaum and Maass [23], and Fowler et al. [15]: again, it is possible to establish upper bounds on approximation factors, and get a factor $\frac{1}{2}$ by a simple greedy approach. Details are omitted from this abstract. We concentrate on the most interesting case of dispersion with boundaries, and only summarize the results for pure dispersion and dispersional packing; it is not hard to see that these variants are related via shrinking or expanding the domain P in an appropriate manner. See the full version of this paper or [2] for details.

The rest of this paper is organized as follows: In Section 3, we show that geometric dispersion with boundaries cannot be approximated arbitrarily well within polynomial time, unless P=NP; this result is valid, even if the polygonal domain has only axis-parallel edges, and distances are measured by the L_∞ metric. In Section 4, we give a strongly polynomial algorithm that approximates this case of geometric dispersion within a factor of $\frac{2}{3}$ of the optimum.

3 An upper bound on approximation factors

In this section, we give a sketch of an NP-completeness proof for geometric dispersion. Basically, we proceed along the lines of Fowler et al. [15], combined with the result by Lichtenstein [17, 25] about the NP-completeness of PLANAR 3SAT. We then argue that our proof implies an upper bound on approximation factors. In this abstract, we concentrate on the case of geometric dispersion with boundaries. In all figures, the boundaries correspond to the original boundaries of P, the interior is shaded in two colors. The lighter one corresponds to the part of the domain that is lost when shrinking P to accommodate for half of the considered distance $L^* = d(v, \partial P)$. The remaining dark region is the part that is feasible for packing $\frac{L^*}{2}$-squares.

Theorem 1. *Unless P=NP, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries, even if the polygonal domain has only axis-parallel edges.*

Sketch: We give a reduction of PLANAR 3SAT. A 3SAT instance I is said to be an instance of PLANAR 3SAT, if the following bipartite graph G_I is planar: Every variable and every clause in I is represented by a vertex in G_I; two vertices are connected, if and only if one of them represents a variable that appears in the clause that is represented by the other vertex. See Figure 1 for an example.

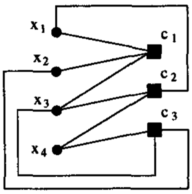

Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$

Proposition 2 (Lichtenstein) PLANAR 3SAT *is NP-complete.*

As a first step, we construct an appropriate planar layout of the graph G_I by using the methods of Duchet et al. [8], or Rosenstiehl and Tarjan [37]. Note that these algorithms produce layouts with all coordinates linear in the number of vertices of G_I.

Next, we proceed to represent variables, clauses, and connecting edges by suitable polygonal pieces. See Figure 2 for the construction of variable components.

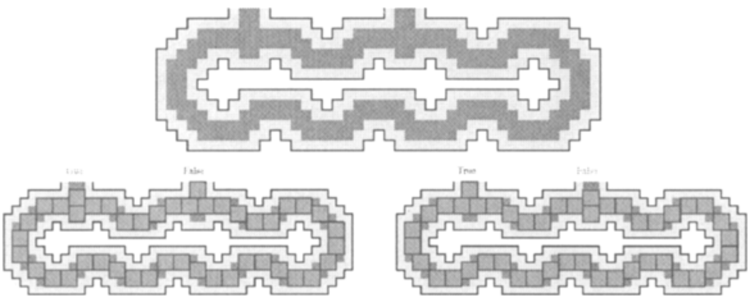

Fig. 2. A variable component for dispersion with boundaries (top), a placement corresponding to "true" (center), and a placement corresponding to "false" (bottom)

The variable components are constructed in a way that allows basically two ways of dispersing a specific number of locations. One of them corresponds to a setting of "true", the other to a setting of "false". Depending on the truth setting, the adjacent connector components will have their squares pushed out or not. See Figure 5 (bottom) for the design of the connector components.

The construction of the clause components is shown in Figure 3: connector components from three variables (each with the appropriate truth setting) meet in such a way that there is a receptor region of "L" shape into which additional squares can be packed. Any literal that does not satisfy the clause forces one of the three corners of the L to be intersected by a square of the connector. Three additional squares can be packed if and only if one corner is not intersected, i. e., if the clause is satisfied.

From the above components, it is straightforward to compute the parameter k, the number of locations that are to be dispersed by a distance of 2. k is polynomial in the number of vertices of G_I and part of the input for the dispersion problem. All vertices of the resulting P have integer coordinates of small size, their number is polynomial in the number of vertices of G_I.

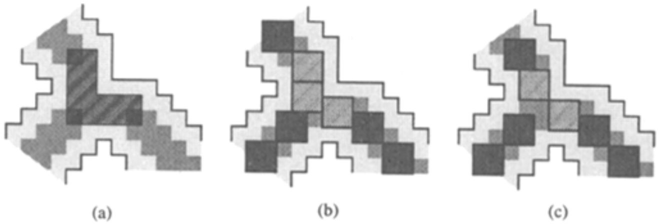

Fig. 3. A clause component for dispersion with boundaries and its receptor region (a), a placement corresponding to "true" (b), and a placement corresponding to "false" (c)

This shows that the problem is NP-hard. Now we need to argue that there cannot be a solution within more than $\frac{13}{14}$ of the optimum, if the PLANAR 3SAT instance cannot be satisfied.

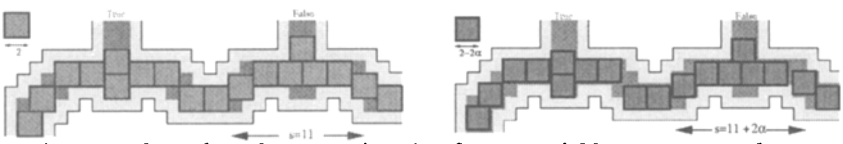

Fig. 4. An upper bound on the approximation factor: variable components for 2-squares (left) and $(2-2\alpha)$-squares (right)

See Figure 4 (top) for the variable components. Indicated is a critical distance of $s = 11$. We show that it is impossible to pack an additional square into this section, even by locally changing the truth setting of a variable. Now suppose there was an approximation factor of $1-\alpha$. This increases the feasible domain for packing squares by a width of 2α, and it decreases the size of these squares to $2-2\alpha$. If $\alpha < \frac{1}{s+3}$, then $\frac{s+1}{2}(2-2\alpha) > s+2\alpha$, implying that it is impossible to place more than $\frac{s+1}{2}$ squares within the indicated part of the component. Similar arguments can be made for all other parts of the construction – see Figure 5. (Details are contained in the full version of the paper.) This shows that it is impossible to make local improvements in the packing to account for unsatisfied clauses, implying that we basically get the same solutions for value $2-2\alpha$ as for value 2, as long as $\alpha < \frac{1}{14}$. □

Along the same lines, we can show the following:

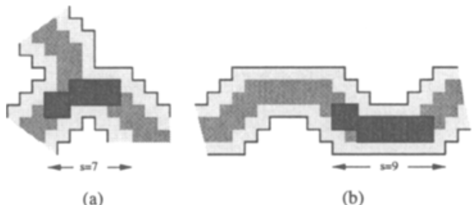

Fig. 5. An upper bound on the approximation factor: clause components (a) and connector components (b)

Theorem 3. *Unless P=NP, there is no polynomial algorithm that can guarantee a solution within more than $\frac{7}{8}$ of the optimum for pure geometric dispersion with L_∞ distances or for dispersional packing, even if the domain has only axis-parallel edges.*

Details are omitted from this abstract. We note that this bound can be lowered to $\frac{1}{2}$ if we do not require P to be a non-degenerate connected domain – see Figure 6 for the general idea. Further technical details are contained in the full version of the paper. It should be noted that the problem of covering a discrete point set, instead of "packing" into it, is well studied in the context of clustering – see the overview by Bern and Eppstein [4].

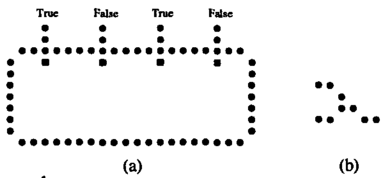

Fig. 6. An upper bound of $\frac{1}{2}$ on the approximation factor, if the domain P may be degenerate and disconnected – variable components (a) and clause components (b)

4 A $\frac{2}{3}$ approximation algorithm

In this section, we describe an approximation algorithm for geometric dispersion with axis-parallel boundaries in the case of L_∞ distances. We show that we can achieve an approximation factor of $\frac{2}{3}$. We use the following notation:

Definition 4 *The* **horizontal α-neighborhood** *of a d square Q is a rectangle of size $((d+\alpha)\times d)$ with the same center as Q.*
For a polygonal domain P and a distance r, $\boldsymbol{P-r}$ *is the polygonal domain $\{p \in P \mid d(p, \partial P) \geq r\}$, obtained by shrinking P. Similarly,* $\boldsymbol{P+r}$ *is the domain $\{p \in \mathbb{R}^2 \mid d(p, P) \leq r\}$, obtained by expanding P. Note that $P+r$ is a polygonal domain for rectilinear distances.*
$\boldsymbol{Par(P)} := \{(e_i, e_j) \mid e_i \,||\, e_j;\ e_i, e_j \in E(P)\}$ *is the set of all pairs of parallel edges of P.* $\boldsymbol{Dist(e_i, e_j)}$ *(for $(e_i, e_j) \in Par(P)$) is the distance of the edges e_i and e_j.*
With $\boldsymbol{AS(P, d, l)}$*, we call the approximation scheme by Hochbaum and Maass*

for $\max_k \text{PACK}(L)$*, where* P *is the feasible domain,* d *is the size of the squares, and* l *is the width of the strips, guaranteeing that the number of packed squares is at least within a factor of* $\left(\frac{l-1}{l}\right)^2$ *of the optimum.*

Note that the approximation scheme $AS(P, d, l)$ can be modified for axis-parallel boundaries, such that the resulting algorithms are strongly polynomial: If the number of squares that can be packed is not polynomial in the number n of vertices of P, then there must be two "long" parallel edges. These can be shortened by cutting out a "large" rectangle, which can be dealt with easily. This procedure can be repeated until all edges are of length polynomially bounded in n. (Details are contained in the full version of the paper. Also, see [2, 40].)

The idea of the algorithm is the following: Use binary search over the size d of the squares in combination with the approximation scheme by Hochbaum and Maas for $\max_k \text{PACK}(L)$ in order to find the largest size d of squares where the approximation scheme guarantees a packing of k many d-squares into the domain $P - \frac{d}{2}$, with the optimum number of d-squares guaranteed to be strictly less than $\frac{3k}{2}$. Then the following crucial lemma guarantees that it is impossible to pack k squares of size at least $\frac{3d}{2}$ into $P - \frac{3d}{4}$, implying a $\frac{2}{3}$ approximation algorithm.

Lemma 5 *Let* P *be a polygonal domain, such that* k *many* $\frac{3d}{2}$*-squares can be packed into* $P-\frac{3d}{4}$*. Then at least* $\frac{3}{2}k$ *many* d*-squares can be packed into* $P-d/2$*.*

Proof. Consider a packing of k many $\frac{3d}{2}$-squares into $P-\frac{3d}{4}$.

Clearly, we have:

$$(P-\frac{3d}{4}) + \frac{d}{4} \subseteq P-d/2. \tag{1}$$

For constructing a packing of d-squares, it suffices to consider the domain that is covered by the $\frac{3d}{2}$-squares instead of $P-\frac{3d}{4}$. After expanding this domain by $\frac{d}{4}$, we get a subset of $P-d/2$ by (1). In the following, we construct a packing of d-squares. At any stage, the following Observation 6 is valid.

Observation 6 *Suppose the feasible space for packing* d*-squares contains the horizontal* $\frac{d}{4}$*-neighborhoods of a set of disjoint* $\frac{3d}{2}$*-squares. Then there exists a* $\frac{3d}{2}$*-square* Q *that has left-most position among all remaining squares, i. e., to the left of* Q*, the horizontal* $\frac{d}{4}$*-neighborhood of* Q *does not overlap the horizontal* $\frac{d}{4}$*-neighborhood of any other* $\frac{3d}{2}$*-square.*

While there are $\frac{3d}{2}$-squares left, consider a leftmost $\frac{3d}{2}$-square Q. We distinguish cases, depending on the relative position of Q with respect to other remaining $\frac{3d}{2}$-squares. See Figure 7.

Details are omitted from this abstract for lack of space. At any stage, a set of one, two, or three $\frac{3}{2}d$-squares that includes a leftmost $\frac{3}{2}d$-square is replaced by a set of two, three, or five d-squares This iteration is performed while there are $\frac{3}{2}d$-squares left. It follows that we can pack at least $\frac{3}{2}k$ many d-squares into $P-\frac{d}{2}$. □

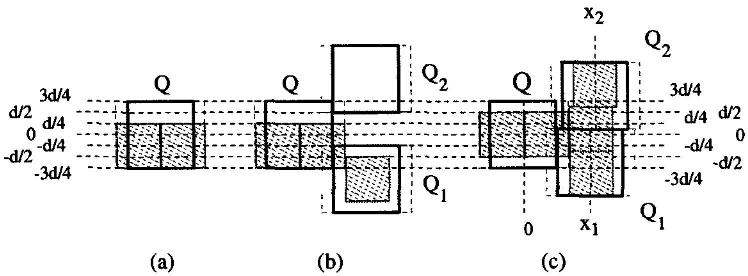

Fig. 7. Constructing a packing of d-squares

In the following, we give a formal description of the binary search algorithm, and argue why it is possible to terminate the binary search in polynomial time.

Algorithm 7

Input: *polygonal domain P, positive integer k.*

Output: $A_{Dis}(P,k) := d$ *is the minimum L_∞ distance between a location and the boundary or between two locations.*

1. **For all** $(e_i, e_j) \in Par(P)$ **do**
2. **While** d_{ij} *undetermined, perform binary search as follows:*
   (a) $s_{\max} := k+1$ *and* $s_{\min} := 2$ *and* $d_{\max} := \frac{2}{3}Dist(e_i, e_j)/(s_{\max})$ *and* $d_{\min} := \frac{2}{3}Dist(e_i, e_j)/(s_{\min})$.
   (b) **If** $AS(P - d_{\max}/2\,,\, d_{\max}\,, 6) < k$ **then** $d_{ij} = 0$.
   (c) **If** $AS(P - d_{\min}/2\,,\, d_{\min}\,, 6) \geq k$ **then** $d_{ij} := \frac{2}{3}Dist(e_i, e_j)$.
   (d) **While** $s_{\max} > s_{\min} + 1$ **do**
   (e) $\quad s := \lceil (s_{max} + s_{min})/2 \rceil$ *and* $d := \frac{2}{3}Dist(e_i, e_j)/s$.
   (f) $\quad$ **If** $AS(P - d/2, d, 6) \geq k$ **then** $s_{max} := s$.
   (g) $\quad$ **Else** $s_{min} := s$.
   (h) $d_{ij} := d$.
3. **Output** $d := \max\{d_{ij} \mid (e_i, e_j) \in Par(P)\}$ *and* **exit**.

Theorem 8. *For rectilinear geometric dispersion with boundaries of k locations in a polygonal domain P with axis-parallel boundaries and n vertices, Algorithm 7 computes a solution $A_{Dis}(P,k)$, such that*

$$A_{Dis}(P,k) \geq \frac{2}{3} OPT(P,k).$$

The running time is strongly polynomial.

Proof. It is not hard to see that there are only finitely many values for the optimal value between the k points. More precisely, we can show that for the optimal distance d_{opt}, the following holds:

There is a pair of edges $(e_i, e_j) \in Par(P)$, such that

$$d_{opt} = \frac{Dist(e_i, e_j)}{s_{ij}} \qquad \text{for some } 2 \leq s_{ij} \leq k+1. \tag{2}$$

In order to determine an optimal solution, we only need to consider values that satisfy Equation (2). For every pair of parallel edges of P, there are only k possible values for an optimal distance of points. Thus, there can be at most $O(n^2k)$ many values that need to be considered.

We proceed to show that the algorithm guarantees an approximation factor of $\frac{2}{3}$.

By binary search, the algorithm determines for every pair of edges $(e_i, e_j) \in Par(P)$ of P a d_{ij} with the following properties:

1. $\frac{3}{2}d_{ij} = Dist(e_i, e_j)/s_{ij}$ $(2 \leq s_{ij} \leq k+1)$ is a possible optimal value for the distance of k points that have to be dispersed in P.
2. Using the approximation scheme [23], at least k many d_{ij}-squares can be packed into $P - d_{ij}/2$, with $d_{ij} = Dist(e_i, e_j)/(s_{ij})$.
3. If $s_{ij} > 2$, then for $\tilde{d_{ij}} := \frac{2}{3}Dist(e_i, e_j)/(s_{ij} - 1)$, we cannot pack k many $\tilde{d_{ij}}$-squares into $P - \tilde{d_{ij}}/2$ with the help of the approximation scheme.

Property 1 follows from (2), Properties 2 and 3 hold as a result of the binary search.

From Lemma 5, we know that at least $\frac{3}{2}k$ many $\frac{2}{3}d_{opt}$-squares can be packed into $P - \frac{1}{3}d_{opt}$, since k many d_{opt}-squares can be packed into $P - d_{opt}/2$.

Let $k_{opt}(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt})$ be the optimal number of $\frac{2}{3}d_{opt}$-squares that can be packed into $P - \frac{1}{3}d_{opt}$. With the parameter $l = 6$, the approximation scheme [23] guarantees an approximation factor of $(\frac{5}{6})^2$. This implies:

$$\begin{aligned} k_{opt}(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt}) &\leq \left(\frac{6}{5}\right)^2 AS(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt}, 6) \\ &< \frac{3}{2}AS(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt}, 6). \end{aligned}$$

It follows that

$$\frac{3}{2}AS(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt}, 6) > k_{opt}(P - \frac{1}{3}d_{opt}, \frac{2}{3}d_{opt}) \geq \frac{3}{2}k$$

This means that at least k squares are packed when the approximation scheme is called with a value of at most $\frac{2}{3}d_{opt}$.

For $\tilde{d_{ij}}$ this means that $\tilde{d_{ij}} > \frac{2}{3}d_{opt}$ and therefore $\frac{3}{2}\tilde{d_{ij}} = Dist(e_i, e_j)/(s_{d_{ij}} + 1) > d_{opt}$.

Hence, for every pair $(e_i, e_j) \in Par(P)$ of edges, the algorithm determines a value d_{ij} that satisfies $\frac{3}{2}d_{ij} = Dist(e_i, e_j)/s_{d_{ij}}$ and is a potential optimal value, and the next larger potential value is strictly larger than the optimal value.

The algorithm returns the d with $d = \max\{d_{ij} \mid (e_i, e_j) \in Par(P)\}$. Therefore, $\frac{3}{2}d \geq d_{opt}$, implying

$$A_{Dis}(P, k) = d \geq \frac{2}{3}\, d_{opt} = \frac{2}{3}OPT(P, k),$$

proving the approximation factor.

The total running time is $O(\log k \cdot n^{40})$. Note that the strongly polynomial modified version of the approximation scheme [23] takes $O(l^2 \cdot n^2 \cdot n^{l^2})$, i.e., $O(n^{38})$ with $l = 6$.

For the case of general polygonal domains, where the boundaries of the domain are not necessarily axis-parallel, Lemma 5 is still valid. In the full version of the paper, we discuss approximation for this more general case.

Without further details, we mention

Theorem 9. *Pure geometric dispersion and dispersional packing can be approximated within a factor of $\frac{1}{2}$ in (strongly) polynomial time.*

These factors can be achieved without use of the approximation scheme via a straightforward greedy strategy; the approximation factor follows from the fact that any packing of k $2d$-balls guarantees a packing of $2k$ many d-balls, and a greedy packing guarantees a $\frac{1}{2}$-approximation for $\max_k \text{PACK}(L)$.

5 Conclusions

We have presented upper and lower bounds for approximating geometric dispersion problems. In the most interesting case of a non-degenerate, connected domain, these bounds still leave a gap; we believe that the upper bounds can be improved. It would be very interesting if some of the lower bounds of $\frac{1}{2}$ could be improved. If we assume that the area of P is large, it is not very hard to see that an optimal solution can be approximated much better. It should be possible to give some quantification along the lines of an asymptotic polynomial time approximation scheme.

It is clear from our above results that similar upper and lower bounds can be established for L_2 distances.

Like for packing problems, there are many possible variants and extensions. One of the interesting special cases arises from considering a *simple polygon*, i. e., a polygonal region without holes. The complexity of this problem is unknown, even if the simple polygon is *rectilinear*, i. e., all its edges are axis-parallel.

Conjecture 10 *The problem* $\text{PACK}(k, L)$ *for* L_infty *distances is polynomial for the class of simple polygons* P.

Acknowledgments

The second author would like to thank Joe Mitchell, Estie Arkin, and Steve Skiena for discussions at the Stony Brook Computational Geometry Problem Seminar [33], which gave some initial motivation for this research.

We thank Jörg Schepers, Joe Mitchell, and the anonymous referees for helpful comments.

References

1. B. S. Baker, D. J. Brown and H. K. Katseff. A 5/4 algorithm for two-dimensional packing, *Journal of Algorithms*, **2**, 1981, 348–368.
2. Christoph Baur. Packungs- und Dispersionsprobleme. Diplomarbeit, Mathematisches Institut, Universität zu Köln, 1997.
3. J. E. Beasley. An exact two–dimensional non-guillotine cutting tree search procedure, *Operations Research*, **33**, 1985, 49–64.
4. M. Bern and D. Eppstein. Clustering. Section 8.5 of the chapter *Approximation algorithms for geometric problems*, in: D. Hochbaum (ed.): Approximation Algorithms for NP-hard Problems. PWS Publishing, 1996.
5. B. K. Bhattacharya and M. E. Houle. Generalized maximum independent set for trees. To appear in: *Journal of Graph Algorithms and Applications*, 1997. mike@cs.newcastle.edu.au.
6. R. Chandrasekaran and A. Daughety. Location on tree networks: p-centre and n-dispersion problems. *Mathematics of Operations Research*, **6**, 1981, 50–57.
7. R. L. Church and R. S. Garfinkel. Locating an obnoxious facility on a network. *Transportation Science*, **12**, 1978, 107–118.
8. P. Duchet, Y. Hamidoune, M. Las Vergnas, and H. Meyniel. Representing a planar graph by vertical lines joining different levels. *Discrete Mathematics*, **46**, 1983, 319–321.
9. E. Erkut. The discrete p–dispersion problem. *European Journal of Operational Research*, **46**, 1990, 48–60.
10. E. Erkut and S. Neumann. Comparison of four models for dispersing facilities. *European Journal of Operations Research*, **40**, 1989, 275–291.
11. S. P. Fekete and J. Schepers. A new exact algorithm for general orthogonal d-dimensional knapsack problems. *Algorithms – ESA '97*, Springer Lecture Notes in Computer Science, vol. 1284, 1997, 144–156.
12. S. P. Fekete and J. Schepers. On more-dimensional packing I: Modeling. Technical report, ZPR 97-288. Available at http://www.zpr.uni-koeln.de/~paper
13. S. P. Fekete and J. Schepers. On more-dimensional packing II: Bounds. Technical report, ZPR 97-289. Available at http://www.zpr.uni-koeln.de/~paper
14. S. P. Fekete and J. Schepers. On more-dimensional packing III: Exact Algorithms. Technical report, ZPR 97-290. Available at http://www.zpr.uni-koeln.de/~paper
15. R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP–complete. *Information Processing Letters*, **12**, 1981, 133–137.
16. Z. Füredi. The densest packing of equal circles into a parallel strip. *Discrete & Computational Geometry*, 1991, 95–106.
17. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the theory of NP–Completeness.* W. H. Freeman and Company, San Francisco, 1979.
18. A. J. Goldmann and P. M. Dearing. Concepts of optimal locations for partially noxious facilities. *Bulletin of the Operational Research Society of America.* **23**, B85, 1975.
19. R. L. Graham and B. D. Lubachevsky. Dense packings of equal disks in an equilateral triangle: from 22 to 34 and beyond. *The Electronic Journal of Combinatorics*, **2**, 1995, #A1.
20. R. L. Graham and B. D. Lubachevsky. Repeated patterns of dense packings of equal disks in a square. *The Electronic Journal of Combinatorics* **3**, 1996, #R16.

21. R. L. Graham, B. D. Lubachevsky, K. J. Nurmela, and P. R. J. Östergård. Dense packings of congruent circles in a circle. Manuscript, 1996. `bdl@research.att.com`.
22. E. Hadjiconstantinou and N. Christofides. An exact algorithm for general, orthogonal, two–dimensional knapsack problems, *European Journal of Operations Research*, **83**, 1995, 39–56.
23. D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, **32**, 1985, 130–136.
24. C. Kenyon and E. Remila, Approximate strip packing. *Proc. of the 37th Annual Symposium on Foundations of Computer Science (FOCS 96)*, 142–154.
25. D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, **11**, 1982, 329–343.
26. Z. Li and V. Milenkovic. A compaction algorithm for non-convex polygons and its application. *Proc. of the Ninth Annual Symposium on Computational Geometry*, 1993, 153–162.
27. B. D. Lubachevsky and R. L. Graham. Curved hexagonal packings of equal disks in a circle. Manuscript. `bdl@research.att.com`.
28. B. D. Lubachevsky, R. L. Graham, and F. H. Stillinger. Patterns and structures in disk packings. 3rd Geometry Festival, Budapest, Hungary, 1996. Manuscript. `bdl@research.att.com`.
29. C. D. Maranas, C. A. Floudas, and P. M. Pardalos. New results in the packing of equal circles in a square. *Discrete Mathematics* **142**, 1995, 287–293.
30. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, **25**, 1995, 59–68.
31. S. Martello and P. Toth, *Knapsack Problems – Algorithms and Computer Implementations*, Wiley, Chichester, 1990.
32. I. D. Moon and A. J. Goldman. Tree location problems with minimum separation. *Transactions of the Institute of Industrial Engineers*, **21**, 1989, 230–240.
33. J. S. B. Mitchell, Y. Lin, E. Arkin, and S. Skiena. *Stony Brook Computational Geometry Problem Seminar.* Manuscript. `jsbm@ams.sunysb.edu`.
34. J. Nelißen. *New approaches to the pallet loading problem.* Technical Report, RWTH Aachen, Lehrstuhl für Angewandte Mathematik, 1993.
35. S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, **42**, 1994, 299–310.
36. D. J. Rosenkrantz, G. K. Tayi, and S. S. Ravi. Capacitated facility dispersion problems. Manuscript, submitted for publication, 1997. `ravi@cs.albany.edu`.
37. P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete & Computational Geometry*, **1**, 1986, 343–353.
38. J. Schepers. Exakte Algorithmen für orthogonale Packungsprobleme. Dissertation, Mathematisches Institut, Universität zu Köln, 1997.
39. A. Tamir. Obnoxious facility location on graphs. *SIAM Journal on Discrete Mathematics*, **4**, 1991, 550–567.
40. M. Wottawa. Struktur und algorithmische Behandlung von praxisorientierten dreidimensionalen Packungsproblemen. Dissertation, Mathematisches Institut, Universität zu Köln, 1996.