Algorithmica (2001) 30: 451–470
DOI: 10.1007/s00453-001-0022-x

Algorithmica
© 2001 Springer-Verlag New York Inc.

Approximation of Geometric Dispersion Problems[1]

C. Baur[2] and S. P. Fekete[3]

Abstract. We consider problems of distributing a number of points within a polygonal region 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 [20] 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. A special case of this observation solves an open problem by Rosenkrantz et al. [31]. 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.

Key Words. Packing, Dispersion, Location problems, Geometric optimization, Bounds on approximation factors.

1. Introduction: Packing and Dispersion Problems. 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 region, 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 region 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 additional sources of difficulties of packing in two dimensions: the shape of the objects may be complicated (see [22] for an example from the clothing industry), or the region of packing may be complicated. In this paper we deal with problems related to packing objects of simple shape (i.e., identical squares) into a polygonal region, which may have holes. It should be noted that even when the structure of regions *and* objects are not complicated, only little is

[1] A preliminary extended abstract version of this paper appears in the proceedings *APPROX* '98 [2]. This work was supported by the German Federal Ministry of Education, Science, Research and Technology (BMBF, Förderkennzeichen 01 IR 411 C7) and is based in part on the first author's thesis [1].

[2] Center for Parallel Computing, Universität zu Köln, D-50923 Köln, Germany. baur@zpr.uni-koeln.de.
[3] Department of Mathematics, TU Berlin, D-10623 Berlin, Germany. fekete@math.tu-berlin.de.

Received October 1, 1998; revised September 17, 1999, and April 17, 2000. Communicated by K. Jansen and J. Rolim. Online publication April 6, 2001.

known—see the papers by Graham, Lubachevsky, and others [13], [17]–[19], [23]–[25] for packing identical disks into a strip, a square, a circle, or an equilateral triangle. Also, see [8] for the problem of packing a maximal number of (not necessarily axis-aligned) squares into a given square. For another tricky variant, see [29] for an overview of the so-called *pallet loading problem*, where we have to pack identical axis-aligned 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. [12]; here and throughout the paper an *L-square* is a rectangle of size $L \times L$, and the set of vertices of a polygonal region includes the vertices of all the holes it may have.

PACK(k, L)
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?

This decision problem is closely related to the following optimization problem:

$\max_k$PACK(L)
Input: A polygonal region 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 [20] that this problem allows a polynomial time approximation scheme: for any fixed m, there is a polynomial-time algorithm that determines a feasible solution within $1 - 1/m$ of the optimum. The idea of the algorithm is to subdivide the packing region by a grid of squares of size sL for an appropriate parameter s. As we will see below, $s = 2m$ is sufficient, implying that s is a constant. Thus, for each of the subregions within a grid square, we can obtain an optimal packing in constant time by enumeration. Furthermore, we can determine the best of s^2 possible solutions that are obtained by s^2 different placements of the $sL \times sL$ grid: for each placement, we shift the grid by a vector (iL, jL) for $i, j = 0, \ldots, s-1$. This is the so-called *shifting strategy*. It is not hard to see that for an optimal placement, any L-square gets intersected in at most $(2s-1)$ subdivisions into grid squares. As a consequence, the total error when summing up the s^2 heuristic solutions obtained from different shiftings is at most $(2s-1)$OPT. This implies that the best of these solutions has cardinality at least the average $((s^2-2s+1)/s^2)\text{OPT} = ((s-1)/s)^2\text{OPT}$. As long as $s \geq m + \sqrt{m(m-1)}$, we get the desired performance guarantee.

The main content of this paper is to examine several versions of the closely related problem

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

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 et al. [31], who give a good overview, including the papers [5], [6], [9], [10], [16], [26], [27], [30], and [33]. In the paper [30], they consider placing k facilities such that the minimum distance between facilities is maximized, and show that a greedy heuristic guarantees a performance ratio of $\frac{1}{2}$ for the problem of maximizing the minimum distance bet. Since their problem is a generalized version of the problem $\max_L \text{PACK}(k)$ defined above, this result applies to all our problems.

All these papers consider discrete sets of possible locations, so the problem can be considered as a generalized independent set problem in a graph. 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 region. We show the close connection to the packing problem and the polynomial approximation scheme by Hochbaum and Maass [20], but also a crucial difference: in general, if P $\neq$ NP, it cannot be expected that the geometric dispersion problem can be approximated arbitrarily well. This resolves an open problem stated in [30]. In particular, we show that the performance guarantee of $\frac{1}{2}$ by [30] is best possible for geometric instances, if the set of feasible facility locations may be disconnected.

Even stronger restrictions apply when we consider geometric instances with the set of feasible locations forming a nondegenerate connected polygonal region. Discussing problems of this type is the main topic of this paper.

In general, a polygonal region may have holes; a *rectilinear polygon* is a polygonal region with all edges axis-parallel.

When placing objects into a polygonal region, 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 is 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 region 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 region P. This version is called *dispersion with boundaries*.

Finally, we may consider a generalization of the problem $\max_L \text{PACK}(k)$, 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*. In all cases, the unit balls that need to be packed arise from some norm, so the orientation is fixed.

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).

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 region P in an appropriate manner.

It is possible to consider various distance functions 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; all ideas carry over for other metrics by combining our ideas with the techniques by Hochbaum and Maass [20] and Fowler et al. [12]. We do not include all the details, but sketch the approach at the end of Section 2.

The main results of this paper are organized as follows: in Section 2 we show that geometric dispersion with boundaries cannot be approximated arbitrarily well within polynomial time, unless P = NP. In Section 3 we give a strongly polynomial algorithm that approximates geometric dispersion within a factor of $\frac{2}{3}$ of the optimum. Other variants and extensions are sketched.

2. An Upper Bound on Approximation Factors. In this section we give an NP-completeness proof for geometric dispersion. We give a reduction of the problem PLANAR 3SAT, which was shown to be NP-complete by Lichtenstein [14], [21]. We proceed along the lines of Fowler et al. [12], who gave a reduction of 3SAT to PACK(k, L); however, we need some extra properties in a reduction to establish an upper bound on approximation factors. We describe the details of our construction for geometric dispersion with boundaries; it is straightforward to see how the result can be extended to the other cases. In all figures the boundaries correspond to the original boundaries of P, the interior is shaded light and dark. The lighter shading corresponds to the part of the region 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 L_∞-balls of size $L^*/2$, i.e., axis-aligned $(L^*/2)$-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.*

PROOF. 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 x_i and every clause c_j 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.

As a first step, we construct a planar rectilinear layout for the graph G_I by using the methods of Duchet et al. [7] and Rosenstiehl and Tarjan [32]. Such a layout draws vertices as horizontal line segments and edges as vertical line segments; any horizontal segment for a vertex v is incident to the vertical segments for precisely the edges adjacent to v. It is a property of these algorithms that they produce layouts with all coordinates being integers that are linear in the number $n + m$ of vertices of G_I, where n is the number of variables and m is the number of clauses; since G_I is planar, we have $m = O(n)$.

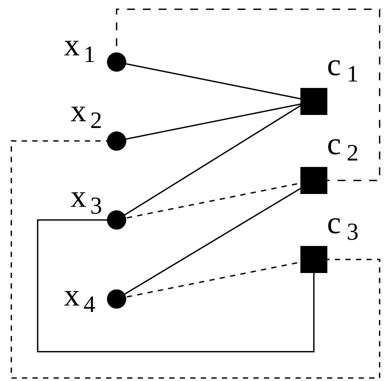

Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. Edges are distinguished according to the logical parity of the corresponding literals.

After scaling this layout by a factor of $O(n^2)$, we replace each vertex representing an element in the set $\{x_i \mid i = 1, \dots, n\}$ of variables by a "variable component," which is a suitable polygonal piece of size $O(n)$. Any such component for a variable v_i that appears in $\delta(v_i)$ clauses has $\delta(v_i)$ pairs of "exits" for attaching possible edges. Any pair corresponds to an edge and has an "even" and an "odd" exit. If the edge corresponds to an unnegated occurrence of the variable in a clause, then the even exit is used, otherwise, the odd exit is used. (In Figure 1 the edges corresponding to unnegated literals are drawn with solid lines, while negated literals are drawn with dashed lines.)

Horizontal segments representing elements in the set $\{v_j \mid j = 1, \dots, m\}$ of clauses are replaced by "clause components" of size $O(1)$. Finally, any of the $3m$ edge segments connecting a variable and a clause is represented by a polygonal piece of size $O(n^3)$, such that, overall, we get one connected polygonal region.

In the following subsections we describe the details of the construction.

2.1. *Variable Components*. See Figure 2 for the construction of the variable component for a variable x_i. (Shown is an example with one pair of exits; variants with more exits are constructed in a similar way by using a periodic replication of the layout.) All polygon edges are axis-parallel, with all vertices having integer coordinates. Adjacent connector components are attached at the positions marked p_t and p_f.

The idea is that a variable component allows basically two ways of dispersing a specific number n_i 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.

We establish these facts by the following lemmas:

LEMMA 2. *For any feasible packing of k squares, there exists a packing of k squares at integer coordinates.*

PROOF. As discussed above, any placement of centers with rectilinear distance 2 from each other and the boundary can be considered as a packing of 2-squares within a

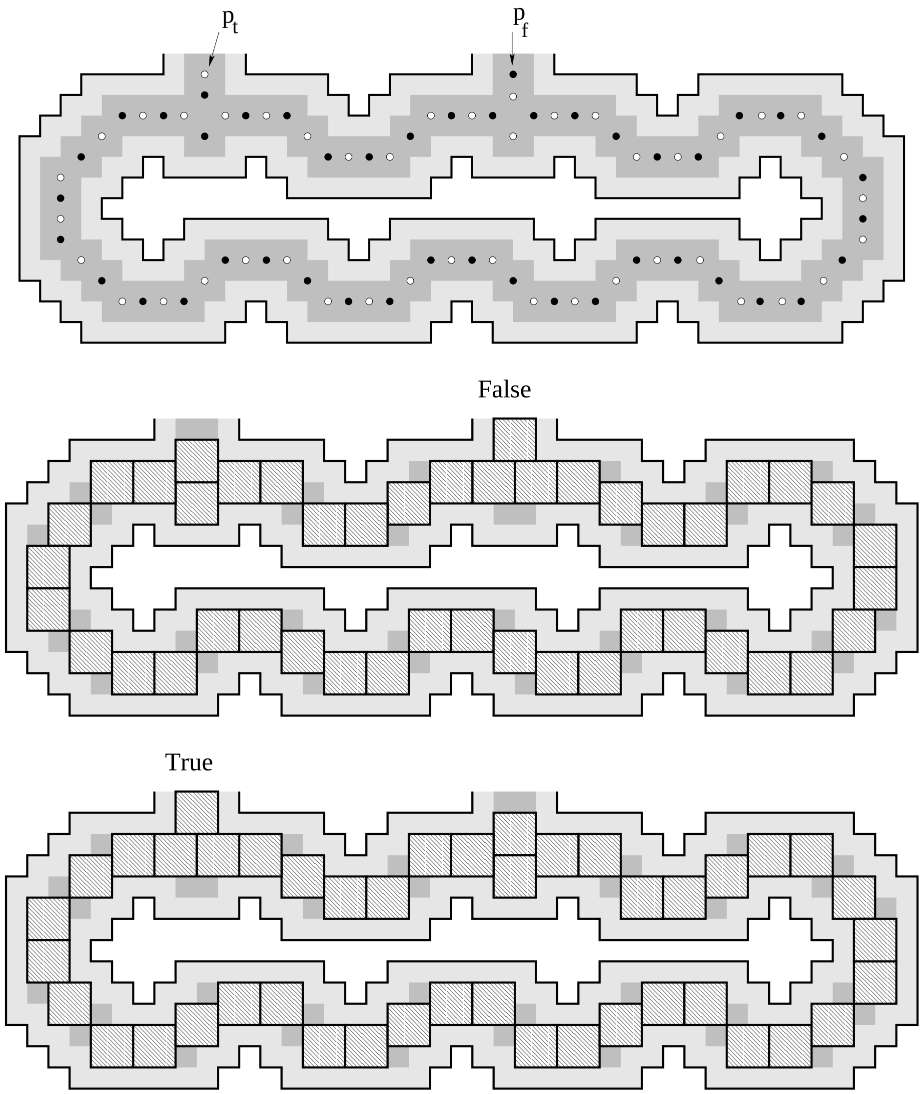

Fig. 2. A variable component for dispersion with boundaries, and the critical points (top), a placement corresponding to "false" (center), and a placement corresponding to "true" (bottom).

shrunk region. Without loss of generality, we can assume that all packings are "lower left justified," i.e., all squares are pushed as far left and down as possible. Since all edges of the polygon are axis-parallel with integer coordinates, and the size of the 2-squares is integral, it follows by an easy induction that all squares must be centered at integer coordinates. □

LEMMA 3. *For each variable component, there are precisely two different lower left justified packings of a maximum number of* 2*-squares*.

PROOF. See Figure 2 (top) and consider the set of $2n_i$ "critical points," which are feasible positions with integer coordinates for a center in a lower left justified packing. These possible locations are subdivided into a set of n_i "black" points and n_i "white" points. If a white (black) point is the center of a square, then no other square can be centered at one of the adjacent black (white) points. This implies that at most n_i squares can be packed into a variable component. Furthermore, if there is a packing of n_i squares, it must consist of only squares centered at black or centered at white points. On the other hand, no two squares located at points of the same color can interfere, showing that we have precisely two feasible sets of n_i 2-squares: one consists of all the white points, the other of all the black points. □

The positions indicated by p_t and p_f correspond to "exits," i.e., positions where adjacent connector components are attached. In one of the two packings, p_t is occupied, corresponding to setting variable x_i to "true." In the other packing, p_f is occupied, corresponding to setting variable x_i to "false."

2.2. *Connector Components*. For each edge connecting the variable to a clause, we place a connector component that links the variable component to the adjacent clause component. Depending on whether the variable appears negated or unnegated in the clause, the connector component is attached to a "true" or "false" exit of the variable component. See Figure 2 for the attachment of connector components to clause components, and Figure 3 for the basic design of the connector components. The connector components follow the edges in the drawing of the graph G_I; since this graph is planar, and the connector components do not interfere at variable components or at clause components, they stay separate. Since the length of the edges in the drawing of G_I resulting from the method by Rosenstiehl and Tarjan is at most $O(n)$, the number of vertices for each connector component is polynomially bounded in n. For reasons that will become obvious in Section 2.4, the connector components do not just represent the (possibly long) straight lines of the edges in the graph, but rather follow them in a zigzagging manner, as shown in Figure 3.

Just like for variable components, we can consider lower left justified packings. It is straightforward to see that each connector component has a maximal number n_e that can be packed into it:

LEMMA 4. *For each connector component, there are precisely two different lower left justified packings of a maximum number n_e of 2-squares.*

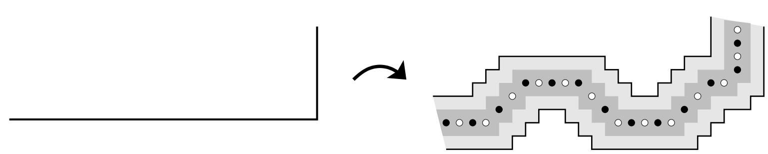

Fig. 3. Part of a connector component (right) for representing an edge between variable nodes and clause nodes (left).

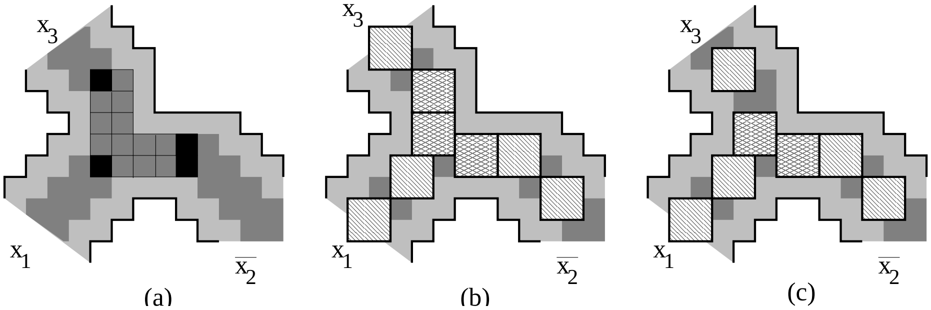

Fig. 4. A clause component for dispersion with boundaries and its receptor region (a); a satisfying placement (b); and an unsatisfying placement (c).

PROOF. Analogous to Lemma 3, we have a set of n_e white points and a set of n_e black points that describe the set of feasible positions in a lower left justified packing. Again, any placement of a 2-square at a white (black) point blocks the adjacent black (white) points, and no two white (black) positions interfere. This implies the claim. □

Moreover, this implies the following:

LEMMA 5. *Any packing of n_e into a connector component must use center points of the same color as the adjacent variable component.*

2.3. *Clause Components*. The construction of the clause components is shown in Figure 4, for the case where literal $\overline{x_2}$ has a different logical parity than x_1, and x_3 has the same parity. (If both literals have the same parity as x_1, the connector components are both attached like x_3; if both literals have the opposite parity, their connector components are both attached like $\overline{x_2}$.)

The connector components from three variables meet in such a way that there is a receptor region of an "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 at least 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: $k = \sum_{i=1}^{n} n_i + \sum_{e \in E} n_e + 3m$.

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. Since there is a packing of k 2-squares, iff there is a satisfying truth assignment for the PLANAR 3SAT instance, we conclude that the problem is NP-hard. We are left to show the claimed bound on the approximation factor.

2.4. *Smaller Distances*. To conclude the proof of Theorem 1, we argue that a solution within more than $\frac{13}{14}$ of the optimum size requires a satisfiable PLANAR 3SAT instance.

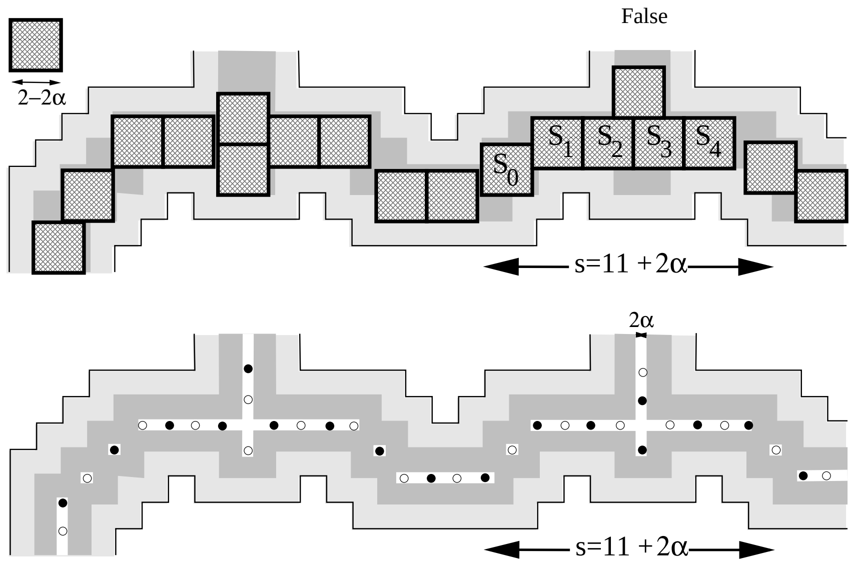

Fig. 5. An upper bound on the approximation factor: variable components for $(2-2\alpha)$-squares (top) and critical points (bottom).

Suppose we have a solution for k many $2(1-\alpha)$-squares, with $\alpha < \frac{1}{14}$. This increases the feasible region for packing squares by a width of α at each boundary, and it decreases the size of the squares to $2-2\alpha$. As before, we assume that this solution is lower left justified. See Figure 5 for a solution of this type in part of a variable component. We argue that there is a corresponding solution for 2-squares, as shown in Figure 6. In both bottom parts of the figures, the sets of feasible center positions are shown as connected white neighborhoods of the critical points. We see that each such neighborhood is the union of a number of horizontal and vertical rectangular strips of width 2α. We say that a strip is a (z_x, z_y)-strip if it contains z_x points with distinct integer x-coordinates, and z_y points with distinct integer y-coordinates. Clearly, any strip is either a $(1, z_y)$-strip or a $(z_x, 1)$-strip.

We say that a $2(1-\alpha)$-square is at x-depth 0 if its left boundary meets the boundary of the feasible region. Inductively, we say that a $2(1-\alpha)$-square is at x-depth i if it is not at x-depth $i-1$ or lower, and its left boundary meets the right boundary of a $2(1-\alpha)$-square at depth $i-1$. The y-depth is defined analogously. By construction, a square at x-level i and y-level j is centered at position $(-(2i+1)\alpha, -(2j+1)\alpha)$, $\mod(1,1)$. Since $\alpha < \frac{1}{14}$, this implies that no $2(1-\alpha)$-square at x-level $1, \ldots, 5$ can be contained in a $(1, z_y)$-strip.

Now consider a $(z_x, 1)$-strip containing a $2(1-\alpha)$-square S_5 at x-level 5. Then there must be a chain of $2(1-\alpha)$-squares $S_4, \ldots, S_0$ at x-level $4, \ldots, 0$, such that the right boundary of S_{i-1} touches the left boundary of S_i. By considering the construction of the components, one sees that all S_i must be centered within the same strip. There is no $(z_x, 1)$-strip with $z_x > 9$; thus, we would have to place $S_0, \ldots, S_5$ within a horizontal distance of not more than a critical distance of $s = 11 + 2\alpha$. (Figure 7 shows the critical

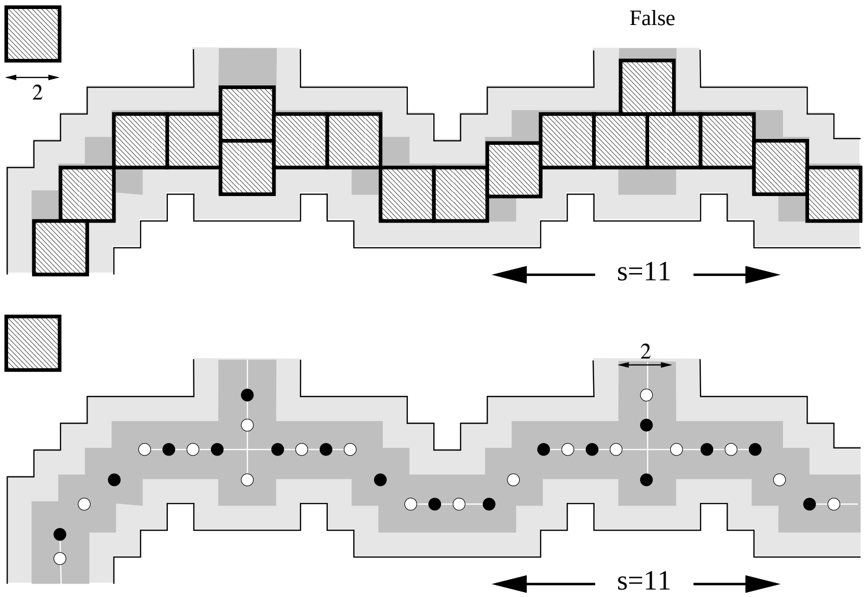

Fig. 6. An upper bound on the approximation factor: variable components for 2-squares (top) and critical points (bottom).

distance for connector components and clause components.) However, $6(2-2\alpha) = 12-12\alpha > 12-\frac{12}{14} = 11+\frac{2}{14} > 11+2\alpha$. Thus, there cannot be a $2(1-\alpha)$-square at level 5. Since there are at least four white and four black points in any $(9, 1)$-strip, we can move each square to the right, until its x-coordinate is integer.

With a similar argument, we conclude that none of the $(5, 1)$-strips that occur in connector components and clause components can contain a $2(1-\alpha)$-square at level 3: then we would have to fit four $2(1-\alpha)$-squares within a distance of $s = 7+2\alpha$, see Figure 7. This is impossible, since $4(2-2\alpha) = 8-8\alpha > 8-\frac{8}{14} = 7+\frac{6}{14} > 7+2\alpha$. As a consequence, we conclude that in any $(5, 1)$-strip, we can move each square to the right, until its x-coordinate is integer.

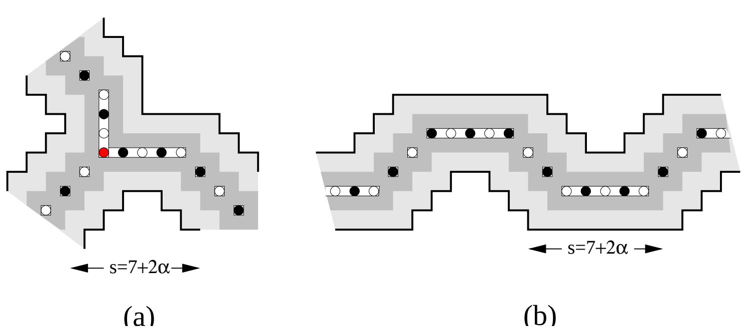

Fig. 7. An upper bound on the approximation factor: clause components (a) and connector components (b).

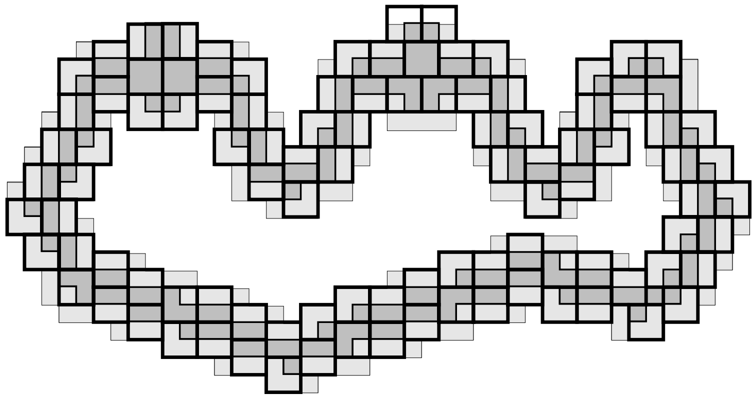

Fig. 8. Pure geometric dispersion: a variable component and a "false" truth assignment.

Similarly, we argue for $(z_x, 1)$-strips with smaller z_x, and for $(1, z_y)$-strips. Thus, we get a solution where all $2(1-\alpha)$-squares are centered at integer points, concluding our proof. □

2.5. *Other Variants*. Using the same construction, we can show similar results for the other forms of dispersion. (See [12] for a basic discussion of the case of unit circles.) In each case we just have to construct a variable component that allows two feasible packings of the same cardinality; a connector component that allows two feasible packings of the same cardinality; and a clause component that allows a packing of three objects if at least one variable satisfies the clause, and only two objects, if none does. As an example, see Figure 8 for a variable component for pure geometric dispersion in nondegenerate connected regions, with a placement corresponding to a "false" truth setting of the variable. We summarize:

THEOREM 6. *Unless P = NP, there is no polynomial-time approximation scheme for pure geometric dispersion or for dispersional packing.*

It has already been pointed out by Fowler et al. [12] that a similar construction can be used for proving hardness of PACK(k, L) for norms other than the L_1 norm. In fact, it is possible to use our approach for establishing the existence of a bound on the approximation factor for *any* norm. See Figure 9 for the case of dispersional packing, where the corresponding unit ball is an ellipse. Shown is a variable gadget with two pairs of exits; the two different shadings for the ellipses ("color classes") indicate the two different feasible placements of a set of maximal cardinality. Connector gadgets are formed like the lower part of the variable gadget. Figure 10 shows a clause gadget. The connections to the three adjacent variable gadgets are from the upper left, the lower left, and the lower right. If one of those three corners of the clause gadget is unoccupied, then two unit balls can be placed. (If the corner on the lower left is unoccupied, then choose the two light unit balls as shown in (a); if one of the two other corners is unoccupied,

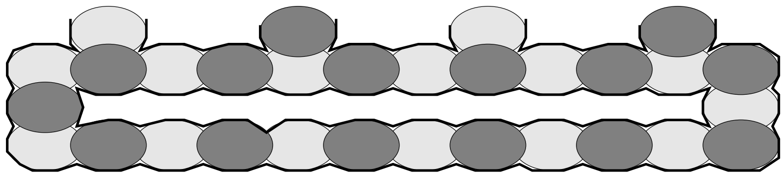

Fig. 9. A variable gadget for the case where the unit ball is an ellipse.

place two disjoint unit balls as shown in (b).) If all three corners are occupied (meaning that none of the corresponding three variables satisfies the clause), then it can be seen from Figure 10(c) that only one additional unit ball can be placed.

Furthermore, it is straightforward to see that the construction can be carried out such that a certain minimal amount of overlap is guaranteed between placements of the same color class, or between critical placements in a clause region. This means that sufficiently small shrinking of the unit balls does not change the overlap of these critical placements. If we interpret this small shrinking as a scaling by a factor sufficiently close to 1, it follows that for this slightly reduced size, we basically get the same solutions as for the original size of the unit balls. Thus, there has to be a bound on the approximation factor, and there cannot be a polynomial-time approximation scheme.

Our above techniques already give a partial answer to an open problem by Ravi et al. [30] by showing a bound on possible approximation factors. In fact, we can modify our techniques to show that the performance ratio of $\frac{1}{2}$ established in [30] is best possible for the case of L_1 or L_∞ distances, if we allow the set of feasible locations to be disconnected. The proof proceeds similar to Theorem 1 by giving a reduction of PLANAR 3SAT. Variable components are chosen as shown in Figure 11(a), consisting of single points at closest L_∞ distance $\frac{1}{2}$. Clause components are chosen as shown in Figure 11(b). If there is a solution with minimum distance 1, no two neighboring points can be chosen, and we must have a satisfying truth assignment. Furthermore, any feasible placement of squares larger than $\frac{1}{2}$ can be transformed into a feasible placement of 1-squares by simply expanding the squares. This means that any approximation algorithm with a performance ratio better than $\frac{1}{2}$ would need to find a satisfying truth assignment, which is impossible, unless $\mathrm{P} = \mathrm{NP}$.

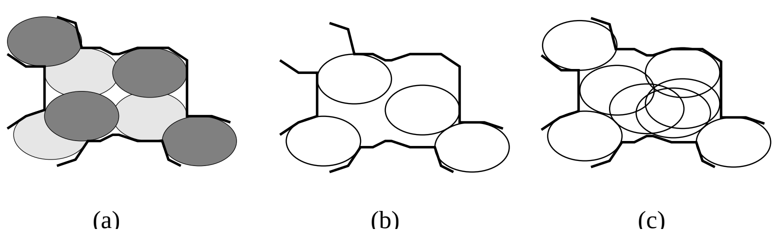

Fig. 10. A clause gadget for the case where the unit ball is an ellipse: placement of ellipses within the clause gadget (a); placing two ellipses when the clause is satisfied (b); placing at most one ellipse when the clause is not satisfied (c).

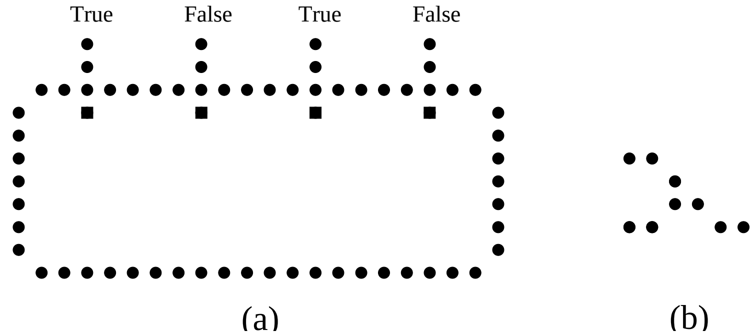

Fig. 11. An upper bound of $\frac{1}{2}$ on the approximation factor, if the region P may be degenerate and disconnected—variable components (a) and clause components (b).

It should be noted that a result of a similar flavor was given by Formann and Wagner [11] for the problem of maximizing the size of nonoverlapping labels in a map. In this problem, the size d is to be maximized subject to the following feasibility condition: for each point p_i in a given finite set P, a square of size d is to be placed with a corner at p_i, such that all squares are disjoint. The decision version of this problem corresponds to the one for a dispersion problem with a discrete set of possible locations. Unless $\mathrm{P} = \mathrm{NP}$, a bound of $\frac{1}{2}$ on a possible approximation factor is proven for this map labeling problem. Despite these similarities, the dispersion problems treated in this paper are a little different: in general, we are dealing with a continuous, connected set of feasible locations instead of a discrete one. This means that other techniques are needed for establishing bounds on approximation factors.

We conclude this section by noting two other types of problems that are loosely related to dispersion problems. Beauquier et al. [3] have studied the problem of tiling finite subpieces of the infinite chessboard with rectangles of fixed size (1, a) and (b, 1); using techniques similar to the one by Fowler et al., they showed that the problem of deciding the existence of a tiling is NP-complete if $a, b \geq 3$, while it is polynomial for $a, b \leq 2$. Finally, 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].

3. A $\frac{2}{3}$ Approximation Algorithm. In this section we describe an approximation algorithm for geometric dispersion with boundaries. We show that we can achieve an approximation factor of $\frac{2}{3}$ for the case where the feasible region is a rectilinear polygon, beating the general performance ratio of $\frac{1}{2}$ that was presented in [30]. We use the following notation:

DEFINITION 7.

- The *α-neighborhood* of a d-square Q is a $(d + \alpha)$-square with the same center as Q.
- The *horizontal α-neighborhood* of a d-square Q is a $((d + \alpha) \times d)$-rectangle with the same center as Q.

- For a polygonal region P and a distance r, $\boldsymbol{P - r}$ is the polygonal region $\{p \in P \mid d(p, \partial P) \geq r\}$ obtained by shrinking P by a distance of r.
- Let $Par(P) := \{(e_i, e_j) \mid e_i \parallel e_j\,;\; e_i, e_j \in E(P)\}$ be the set of all pairs of parallel edges of P.
- Let $Dist(e_i, e_j)$ (for $(e_i, e_j) \in Par(P)$) be the distance of the edges e_i and e_j.
- With $AS(P, d, l)$, we call the approximation scheme from [20] for the problem $\max_k$ PACK(L), where P is the feasible region, 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 $((l-1)/l)^2$ of the optimum, as described in the Introduction.

Note that the approximation scheme $AS(P, d, l)$ can be modified, 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 that can be packed optimally. (In terms of the original "shifting technique" by Hochbaum and Maass, this corresponds to cutting the rectangle only twice at a large distance, instead of slicing it up into an exponential number of parallel strips.) This procedure can be repeated until all edges are of length polynomially bounded in n.

In the following, we describe our algorithm. The idea is to perform binary search over the set of possible distances d between a location and the boundary, or between two locations. At each step, we try to find a solution of distance d for k locations by calling the approximation scheme $AS(P - d/2, d, l)$ for the number of locations at distance d. The binary search determines the largest distance d for which the approximation scheme with $l = 6$ finds a feasible solution for k locations. The approximation parameter l is fixed to $l = 6$, the smallest integer for which the approximation factor $((l-1)/l)^2$ of $AS(P - d/2, d, l)$ is larger than $\frac{2}{3}$, because then the approximation scheme guarantees that there is no feasible placement of more than 36/25*k many d-squares. As we will see from Lemma 10, this implies that there cannot be a solution for k many $(3d/2)$-squares.

We first describe a simple version of the binary search algorithm that works for rectilinear polygons P. After proving its correctness and performance ratio, we sketch how the binary search can be performed more efficiently. The case of general polygons P is addressed in the concluding section.

ALGORITHM 8.
Input: rectilinear polygon P, positive integer k.
Output: a set of k locations, such that $A_{Dis}(P, k) := d$ 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**
   (a) Perform binary search for the smallest integer m, $2 \leq m \leq k + 1$, with the following property:
      - For $d_{ijm} := Dist(e_i, e_j)/m$, $AS(P - d_{ijm}/2, d_{ijm}, 6)$ returns a feasible solution for at least k locations at distance d_{ijm}.

   (b) Let d_{ij} be the distance d_{ijm} for the critical value m.
2. Let d be the maximum d_{ij} for any (e_i, e_j).

THEOREM 9. *For rectilinear geometric dispersion with boundaries of k locations in a rectilinear polygon P with n vertices, Algorithm* 8 *computes a solution* $A_{Dis}(P, k)$, *such that*

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

The running time is strongly polynomial.

For proving this theorem, we need the following lemma:

LEMMA 10. *Let P be a rectilinear polygon such that k many* $(3d/2)$*-squares can be packed into* $P - 3d/4$. *Then at least* $\frac{3}{2}k$ *many d-squares can be packed into* $P - d/2$.

PROOF. Consider a packing of k many $(3d/2)$-squares into $P - 3d/4$.

Clearly, we have

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

For constructing a packing of d-squares, it suffices to consider the region that is covered by the $(3d/2)$-squares instead of $P - 3d/4$. After expanding this region by $d/4$, 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 is valid.

OBSERVATION 11. *Suppose the feasible space for packing d-squares contains the horizontal* $(d/4)$*-neighborhoods of a set of disjoint* $(3d/2)$*-squares. Then there exists a* $(3d/2)$*-square Q that has the leftmost position among all remaining squares, i.e., to the left of Q, the horizontal* $(d/4)$*-neighborhood of Q does not overlap the horizontal* $(d/4)$*-neighborhood of any other* $(3d/2)$*-square.*

While there are $(3d/2)$-squares left, consider a leftmost $(3d/2)$-square Q. We distinguish cases, depending on the relative position of Q with respect to other remaining $(3d/2)$-squares. See Figure 12. The positions of squares are described by their central points, we assume that Q has position $(0, 0)$.

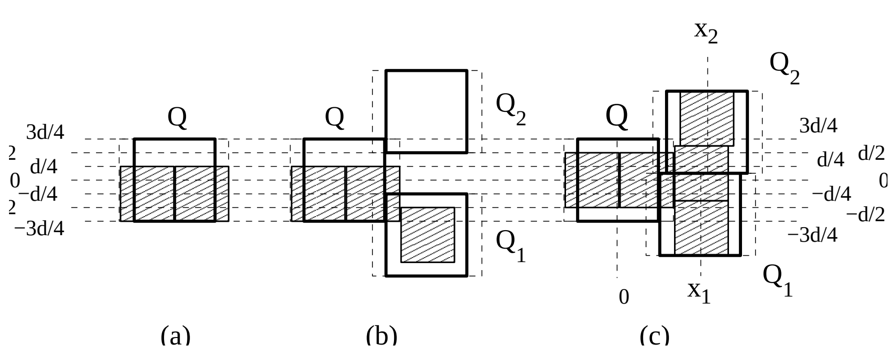

Fig. 12. Constructing a packing of d-squares.

1. To the right of Q, the horizontal $(d/4)$-neighborhood overlaps no horizontal $(d/4)$-neighborhood of another $(3d/2)$-square. See Figure 12(a). Hence we can pack two d-squares into the horizontal $(d/4)$-neighborhood of Q, at positions $(-d/2, -d/4)$ and $(d/2, -d/4)$ without intersecting the horizontal $(d/4)$-neighborhood of any other $(3d/2)$-square.

   After removing Q from the packing of $(3d/2)$-squares, the assumption of Observation 11 is still valid for the $k-1$ remaining $(3d/2)$-squares.
2. To the right of Q, the horizontal $(d/4)$-neighborhood of Q overlaps the horizontal $(d/4)$-neighborhood of one or two other $(3d/2)$-squares. Without loss of generality, consider the case of two $(3d/2)$-squares Q_1 and Q_2, centered at (x_1, y_1) and (x_2, y_2), and assume that $y_1 < y_2$ and $|y_1| \leq |y_2|$, implying $y_1 < 0 < y_2$. Consider the cases:
   (a) $y_2 > d$. See Figure 12(b). Pack two d-squares into the horizontal $(d/4)$-neighborhood of Q like in Case 1, positioned at $(-d/2, -d/4)$ and $(d/2, -d/4)$, with the second one possibly intersecting the $(d/4)$-neighborhood of Q_1, but no $(d/4)$-neighborhood of one of the other $k-2$ $(3d/2)$-squares. Since $x_1 \geq 3d/2$, a third d-square can be placed at (x_1, y_1), so it does not intersect the horizontal $(d/4)$-neighborhood of any other $(3d/2)$-square.

      Packing these three d-squares and removing both Q and Q_1 from the packing of $(3d/2)$-squares leaves the assumption of Observation 11 intact for the remaining feasible space.
   (b) $y_2 \leq d$. See Figure 12(c). Then $y_2 - y_1 < 2d$, so the $(d/4)$-neighborhoods of Q_1 and Q_2 do overlap, and a d-square P_0 placed at $(3d/2, y_1 + 3d/4)$ stays within the union of these neighborhoods. Since Q_1 and Q_2 are disjoint, we must have $y_2 - y_1 \geq 3d/2$, so $y_1 \leq -d/2$ and $d/2 \leq y_2$. It is not hard to see that as a consequence, P_0 can only intersect the horizontal $(d/4)$-neighborhoods of Q, Q_1, Q_2. Furthermore, we can place two more d-squares P_1 and P_2 at positions $(x_1, y_1 - d/4)$ and $(x_2, y_2 - d/4)$, without intersecting horizontal $(d/4)$-neighborhoods other than the ones of Q, Q_1, Q_2. Finally, we pack d-squares P_3 and P_4 into the horizontal $(d/4)$-neighborhood of Q at positions $(-d/2, 0)$ and $(d/2, 0)$. Since $x_1 \geq 3d/2$ and $x_2 \geq 3d/2$, the interior of P_3 and P_4 remains disjoint from the interior of P_0, P_1, and P_2.

      After removing Q, Q_1, Q_2 from the set of $(3d/2)$-squares, the assumptions of Observation 11 are still valid.

This iteration is performed while there are $(3d/2)$-squares left. Since at any stage, we replace a set of i many $(3d/2)$-squares by at least $3i/2$ many d-squares, it follows that we can pack at least $3k/2$ many d-squares into $P - d/2$. □

PROOF OF THEOREM 9. It is not hard to see that there are only finitely many values for the optimal value between the k points. More precisely, the following holds for the optimal distance d_{opt}:

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

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

In order to determine an optimal solution, we only need to consider values that satisfy (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, an m with the following properties:

1. $\frac{3}{2}d_{ij} = Dist(e_i, e_j)/m$ $(2 \le m \le k+1)$ is a possible optimal value for the distance of k points that have to be dispersed in P.
2. Using the approximation scheme [20], at least k many d_{ij}-squares can be packed into $P - d_{ij}/2$, with $d_{ij} = Dist(e_i, e_j)/m$.
3. If $m > 2$, then, for $\tilde{d_{ij}} := \frac{2}{3}Dist(e_i, e_j)/(m-1)$, 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 10, we know that at least $3k/2$ many $\frac{2}{3}d_{\text{opt}}$-squares can be packed into $P - \frac{1}{3}d_{\text{opt}}$, since k many d_{opt}-squares can be packed into $P - d_{\text{opt}}/2$.

Let $k_{\text{opt}}(P - \frac{1}{3}d_{\text{opt}}, \frac{2}{3}d_{\text{opt}})$ be the optimal number of $(2d_{\text{opt}}/3)$-squares that can be packed into $P - \frac{1}{3}d_{\text{opt}}$. With the parameter $l = 6$, the approximation scheme [20] guarantees an approximation factor of $(\frac{5}{6})^2$. This implies

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

It follows that

$$\tfrac{3}{2}AS(P - \tfrac{1}{3}d_{\text{opt}}, \tfrac{2}{3}d_{\text{opt}}, 6) > k_{\text{opt}}(P - \tfrac{1}{3}d_{\text{opt}}, \tfrac{2}{3}d_{\text{opt}}) \ge \tfrac{3}{2}k.$$

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

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

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, $\frac{3}{2}d \ge d_{\text{opt}}$, implying

$$A_{Dis}(P, k) = d \ge \tfrac{2}{3}\, d_{\text{opt}} = \tfrac{2}{3}OPT(P, k),$$

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 [20] takes $O(l^2 \cdot n^2 \cdot n^{l^2})$, i.e., $O(n^{38})$ with $l = 6$. □

It should be noted that it is possible to speed up Algorithm 8 by better organization of the binary search. Instead of performing binary search for all $O(n^2)$ pairs (e_i, e_j), the $O(n^2)$ values $Dist(e_i, e_j)$ can be sorted in time $O(n^2 \log n)$; then the $O(n^2k)$ values $Dist(e_i, e_j)/m$ form an n^2 by k matrix with nondecreasing rows and columns. Using sorted matrix techniques (as described in [15] for geometric problems), it is possible to find the optimal d with $O(\log n + \log k)$ calls to the approximation scheme, and $O(n^2 + \log k)$ additional overhead. As the exponent of n in the resulting complexity is still a rather large 38, we leave it to the interested reader to work out the details.

4. Conclusions. We have presented upper and lower bounds for approximating geometric dispersion problems. In the most interesting case of a nondegenerate, connected region, 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 than within a factor of $\frac{1}{2}$ or even $\frac{2}{3}$. It should be possible to give some quantification along the lines of an asymptotic polynomial-time approximation scheme.

We believe that it is possible to modify Algorithm 8 to make it work for the case of general polygonal regions. Lemma 10 still applies, so we only need to argue that similar to the condition in (2), there is a discrete set of critical sizes for d that can be examined by binary search, such that the resulting complexity is polynomial in $O(\log k)$. It is clear that any critical size d_{ij} is determined by a set of at most k squares that cannot be expanded locally. For rectilinear polygons we saw that this corresponds to a "row" or "column" of at most k squares that fill the distance between two parallel polygon edges. For general polygonal regions, it is not too hard to obtain more general conditions on locally maximal sets, but it takes some extra work to get sufficient structure in the resulting set of critical sizes to be able to apply matrix search techniques, and the resulting algorithm is strongly polynomial.

As we sketched above, similar upper and lower bounds can be established for other norms. It should be interesting to tighten some of the resulting approximation bounds.

Like for packing problems, there are many possible variants and extensions. One of the interesting special cases arises from considering a *simple* connected region. The complexity of this problem is unknown.

CONJECTURE 12. *The problem* PACK(k, L) *is polynomial for the class of simple rectilinear polygons* P.

It is clear that any locally optimal solution can be assumed to be lower left justified, i.e., the squares can be packed in some greedy fashion from the lower left; however, at any stage, there may be many possible greedy placements for a new square. Choosing a lexicographic minimum of the feasible positions may not yield the optimum, as can be seen from Figure 13.

Acknowledgments. The second author thanks Joe Mitchell, Estie Arkin, and Steve Skiena for discussions at the Stony Brook Computational Geometry Problem Seminar [28], which gave some initial motivation for this research. We thank Jörg Schepers and four anonymous referees for helpful comments that improved the presentation of this paper.

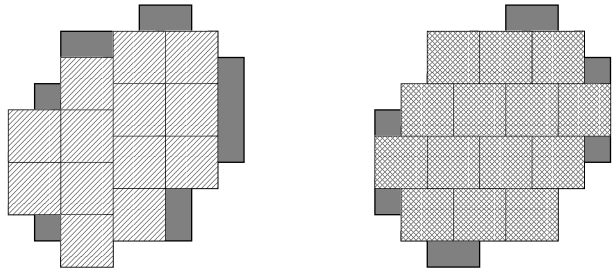

Fig. 13. An instance with a best lexicographic greedy packing (left); the optimal packing (right).

References

[1] C. Baur. Packungs- und Dispersionsprobleme. Diploma thesis, Mathematisches Institut, Universität zu Köln, 1997.

[2] C. Baur and S. P. Fekete. Approximation of geometric dispersion problems (extended abstract). *Approximation Algorithms for Combinatorial Optimization – APPROX '98*. Lecture Notes in Computer Science, vol. 1444. Springer-Verlag, Berlin, 1998, pp. 63–75.

[3] D. Beauquier, M. Nivat, E. Remila, and M. Robson. Tiling figures of the plane with two bars. *Computational Geometry*: *Theory and Applications*, **5** (1995), 1–25.

[4] M. Bern and D. Eppstein. Clustering. In D. Hochbaum (ed.): *Approximation Algorithms for NP-Hard Problems*. PWS, Boston, MA, 1996, Section 8.5 of the chapter Approximation algorithms for geometric problems, pp. 325–329.

[5] R. Chandrasekaran and A. Daughety. Location on tree networks: p-centre and n-dispersion problems. *Mathematics of Operations Research*, **6** (1981), 50–57.

[6] R. L. Church and R. S. Garfinkel. Locating an obnoxious facility on a network. *Transportation Science*, **12** (1978), 107–118.

[7] 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.

[8] P. Erdős and R. L. Graham. On packing squares with equal squares. *Journal of Combinatorial Theory, Series A*, **19** (1975), 119–123.

[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] M. Formann and F. Wagner. A packing problem with applications to lettering of maps. *Proceedings of the Seventh Annual Symposium on Computational Geometry*, 1991, pp. 281–288.

[12] 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.

[13] Z. Füredi. The densest packing of equal circles into a parallel strip. *Discrete & Computational Geometry*, **6** (1991), 95–106.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[15] A. Glozman, K. Kedem, and G. Shpitalnik. On some geometric selection and optimization problems via sorted matrices. *Computational Geometry*: *Theory and Applications*, **11** (1998), 17–28.

[16] A. J. Goldmann and P. M. Dearing. Concepts of optimal locations for partially noxious facilities (short abstract). *Bulletin of the Operational Research Society of America*, **23** (1975), B85.

[17] 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.

[18] 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.

[19] R. L. Graham, B. D. Lubachevsky, K. J. Nurmela, and P. R. J. Östergård. Dense packings of congruent circles in a circle. *Discrete Mathematics*, **181** (1998), 139–154.

[20] 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.

[21] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, **11** (1982), 329–343.

[22] Z. Li and V. Milenkovic. A compaction algorithm for non-convex polygons and its application. *Proceedings of the Ninth Annual Symposium on Computational Geometry*, 1993, pp. 153–162.

[23] B. D. Lubachevsky and R. L. Graham. Curved hexagonal packings of equal disks in a circle. *Discrete & Computational Geometry*, **18** (1997), 179–194.

[24] B. D. Lubachevsky, R. L. Graham, and F. H. Stillinger. Patterns and structures in disk packings. *Periodica Mathematica Hungarica*, **34** (1997), 123–143.

[25] 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.

[26] 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.

[27] I. D. Moon and A. J. Goldman. Tree location problems with minimum separation. *Transactions of the Institute of Industrial Engineers*, **21** (1989), 230–240.

[28] J. S. B. Mitchell, Y. Lin, E. Arkin, and S. Skiena. Stony Brook Computational Geometry Problem Seminar. Manuscript, 1993. `jsbm@ams.sunysb.edu`.

[29] J. Nelißen. New approaches to the pallet loading problem. Technical Report, Lehrstuhl für Angewandte Mathematik, RWTH Aachen, 1993.

[30] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, **42** (1994), 299–310.

[31] D. J. Rosenkrantz, G. K. Tayi, and S. S. Ravi. Facility dispersion problems under capacity and cost constraints. *Journal of Combinatorial Optimization*, **4** (2000), 7–33.

[32] P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete & Computational Geometry*, **1** (1986), 343–353.

[33] A. Tamir. Obnoxious facility location on graphs. *SIAM Journal on Discrete Mathematics*, **4** (1991), 550–567.