

# Hardness of Approximate Compaction for Nonplanar Orthogonal Graph Drawings

Michael J. Bannister and David Eppstein

Computer Science Department, University of California, Irvine

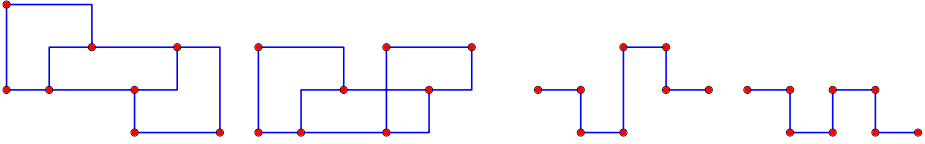
**Abstract.** We show that several problems of compacting orthogonal graph drawings to use the minimum number of rows or the minimum possible area cannot be approximated to within better than a polynomial factor in polynomial time unless  $P = NP$ . However, there is a fixed-parameter-tractable algorithm for testing whether a drawing can be compacted to a given number of rows.

## 1 Introduction

*Orthogonal graph drawing* is a widely used graph drawing style for low-degree graphs, in which each vertex is represented as a point or a rectangle in an integer grid, and each edge is represented as a polyline composed out of axis-parallel line segments [4]. When used for nonplanar graphs, orthogonal drawing has several desirable properties including polynomial area, high angular resolution, and right-angled edge crossings; the last property, in particular, has been shown to aid in legibility of graph drawings [6].

Typical orthogonal graph drawing systems employ a multiphase approach [1,4] in which the input graph is *planarized* by replacing its crossings with vertices, a *topological embedding* of the graph (specifying the ordering of the edges around each vertex, but not the vertex and edge locations) is found, a flow algorithm is used to orient the edges in a way that minimizes the number of bends [10], and vertex coordinates are assigned. If vertices of degree greater than four exist, they may be expanded to rectangles as another phase of this process [1]. Finally, the drawing is improved by *compaction*, a step in which the vertices and bends of the graph are moved to new locations in order to reduce the area of the drawing while preserving its edge orientations and other features.

Some positive algorithmic results are known for the final compaction step; for instance, Bridgeman et al. [2] showed that planar orthogonal drawings in which the shapes of the faces in the drawing are restricted (so-called *turn-regular drawings*) may be compacted into optimal area in polynomial time. However, when drawing nonplanar graphs, it may not be necessary or desirable for the compaction phase to preserve a fixed planarization of the graph. If one is compacting one dimension of a drawing at a time, then for planar compaction it is only possible to map the rows of the drawing monotonically to a smaller set of rows, while for nonplanar graphs it may also be useful to permute the rows with respect to each other. This greater freedom to choose how to compact the drawing



**Fig. 1.** Left: input and output drawings for row-by-row compaction. Right: input and output drawings for vertex-by-vertex compaction.

may lead to much greater savings in drawing area, but it also leads to greater difficulty in finding a good compaction.

As Patrignani [9] showed, even for arbitrary planar orthogonal graph drawings, compacting the drawing in a way that minimizes its area, total edge length, or maximum edge length is NP-hard. Although these results do not directly extend to the nonplanar case, NP-hardness in that case also follows from results of Eades et al. on rectilinear (bendless) drawing [3], and Mañuch et al. where certain restricted cases of rectilinear drawing are considered [8]. But since compaction is performed primarily for aesthetic reasons (a smaller area drawing allows the drawing to be viewed at a larger scale, making its features more legible), exact optimization may not be important as long as a layout with small area can be achieved. Thus, we are led to the problem of how closely it is possible to *approximate* the minimum area layout. The problem of approximate compaction for nonplanar orthogonal drawings was explicitly listed as open by Eiglsperger et al. [4], and there appears to have been little progress on it since then.

In this paper we show that nonplanar compaction is hard even to approximate: there exists a real number  $c > 0$  such that, unless  $P = NP$ , no polynomial time algorithm can find a compaction of a drawing with  $n$  features that is within a factor of  $n^c$  of optimal. The main idea is to find approximation-preserving reductions from graph coloring, a problem known to be hard to approximate. We also find *fixed-parameter tractable* algorithms for finding compactions that use very small numbers of grid rows, for drawings for which such a compaction is possible.

## 1.1 Variations of the Compaction Problem

In the compaction problems we study, the task is to move vertices and bends while preserving the axis-parallel orientation (although not necessarily the direction) of each edge, to minimize the number of rows or area of the drawing. Our results apply either to *orthogonal drawings* (drawings in which edges may be polylines with bends, possible for any graph of maximum degree four) or *rectilinear drawings* (bendless drawings, only possible for some graphs) [3,5]: the distinction between bends and vertices is unimportant for our results.

We distinguish between three variants of the compaction problem, depending on what vertex motions are allowed. In *row-by-row compaction* (Figure 1, left), the compacted layout maps each row of the input layout to a row of the output; all vertices that belong to the same row must move in tandem. In *vertex-*

*by-vertex vertical compaction* (Figure 1, right), each vertex or bend may move independently, but only its  $y$ -coordinate may change; it must retain its horizontal position. In *vertex-by-vertex free compaction*, vertices or bends may move arbitrarily in both coordinate directions. In all three of these problems, edges or edge segments must stay vertical or horizontal according to their orientation in the original layout. The compaction is not allowed to cause any new intersection between a vertex and a feature it was not already incident with, nor is it allowed to cause any two edges or edge segments to overlap for nonzero length; however, it may introduce new crossings that were not previously present.

## 1.2 New Results

We show the following results.

- In the row-by-row compaction problem, it is difficult to compact even a drawing of a path graph (or a drawing of the two-vertex graph with many bends): if the drawing has  $n$  vertices or bends, then unless  $P = NP$  there is no polynomial time algorithm that can find a compacted drawing whose number of rows is within  $O(n^{1/2-\epsilon})$  of optimal, or whose area is within  $O(n^{1/2-\epsilon})$  of optimal, for any  $\epsilon > 0$ . Moreover, even finding drawings with a fixed number of rows is hard: it is  $NP$ -complete to determine whether there exists a compaction with only three rows.
- In vertex-by-vertex vertical compaction, there exist orthogonal graph drawings of maximum degree three such that, unless  $P = NP$ , there is no polynomial time algorithm that can find a compacted drawing whose number of rows is within  $O(n^{1/4-\epsilon})$  of optimal, or whose area is within  $O(n^{1/4-\epsilon})$  of optimal, where  $n$  is the number of features in the drawing, for any  $\epsilon > 0$ . The same result also applies in the vertex-by-vertex free compaction problem.
- For vertex-by-vertex vertical or free compaction of three-dimensional orthogonal drawings, it is not possible (unless  $P = NP$ ) to approximate the minimum number of layers in any one dimension to within  $O(n^{1/2-\epsilon})$  of optimal in polynomial time, for any  $\epsilon > 0$ , nor is it possible in polynomial time to determine whether a three-layer drawing exists.
- In row-by-row and vertex-by-vertex vertical compaction in either two or three dimensions, there is an approximation algorithm with approximation ratio  $O(\sqrt{n})$ , showing that some of our inapproximability bounds are tight.
- In vertex-by-vertex vertical compaction, there is an algorithm for testing whether an orthogonal graph drawing can be compacted into  $k$  rows, whose running time is  $O(k!n)$ . Thus, the problem is fixed-parameter tractable.

## 2 Preliminaries

### 2.1 Orthogonal Drawing

We define an *orthogonal drawing* of a graph to be a drawing in which each vertex is represented as a point in the Euclidean plane (although most of our

results apply as well to drawings in which the vertices are rectangles), and each edge is represented as a polyline (a polygonal chain of line segments), with each line segment parallel to one of the coordinate axes. If each edge is itself a line segment, the drawing is *rectilinear*; otherwise, the segments of a polyline meet at *bends*. Each vertex or bend must only intersect the edges that it belongs to, and no two vertices or bends may coincide. Edges may cross each other, but only at right angles, at points that are neither vertices nor bends.

It is natural, in orthogonal drawing, to restrict the coordinates of the vertices and bends to be integers. In this case, the *width* of a two-dimensional drawing is the maximum difference between the  $x$ -coordinates of any two of its vertices or bends, the *height* is the maximum difference between  $y$ -coordinates of any two vertices or bends, and the *area* is the product of the width and height.

A *compaction* of a drawing  $D$  is another drawing  $D'$  of the same graph, in which the vertices and bends of  $D'$  correspond one-for-one with the vertices and bends of  $D$ , and in which corresponding segments of the two drawings are parallel to each other. Typically,  $D'$  will have smaller height or area than  $D$ . We distinguish between three types of compaction:

- In *row-by-row compaction*, the  $x$ -coordinate of each vertex or bend remains unchanged, and two vertices or bends that have the same  $y$ -coordinate in  $D$  must continue to have the same  $y$ -coordinate in  $D'$  (Figure 1, left).
- In *vertex-by-vertex vertical compaction*, the  $x$ -coordinate of each vertex or bend remains unchanged, but the  $y$ -coordinates may vary independently of each other subject to the condition that the result remains a valid drawing with edge segments parallel to the original drawing (Figure 1, right).
- In *vertex-by-vertex free compaction*, the  $x$ - and  $y$ -coordinates of each vertex or bend are free to vary independently of other vertices or bends.

As can be seen in Figure 1, we allow compaction to introduce new edge crossings and to reverse the directions of edge segments. These concepts generalize straightforwardly to three dimensions.

## 2.2 Graph Coloring and Inapproximability

In the *graph coloring problem*, we are given as input a graph and seek to color the vertices of the graph with as few colors as possible, in such a way that the endpoints of each edge are assigned different colors. Our results on the difficulty of compaction are based on known inapproximability results for graph coloring, one of the triumphs of the theory of probabilistically checkable proofs.

**Lemma 1 (Zuckerman [11]).** *Let  $\epsilon > 0$  be any fixed constant. Then, unless  $P = NP$ , there is no polynomial time algorithm that can color a given  $n$ -vertex graph using a number of colors within a factor of  $n^{1-\epsilon}$  of the optimal number.*

Our proofs use approximation-preserving reductions from coloring to compaction: given a graph  $G$  to be colored, we will construct a different graph  $G'$  and a drawing  $D$  of  $G'$  such that the layers in a compaction  $D'$  of  $D$  necessarily correspond

to the colors in a coloring of  $G$ . With a reduction of this type, the approximation ratio for compacting  $D$  cannot be better than the approximation ratio for coloring  $G$ . However,  $D$  will in general have many more vertices and bends than the number of vertices in  $G$ : the size of  $D$  will be at least proportional to the number of edges in  $G$ , which is quadratic in its number of vertices. Therefore, although the approximation ratio will remain unchanged as a number by our reduction it will be expressed as a different function of the input size.

### 2.3 Notation

We write  $n_G$ ,  $n_D$ , or (where unambiguous)  $n$  for the number of vertices in a graph  $G$  or drawing  $D$  and  $m_G$ ,  $m_D$ , or  $m$  for its number of edges. Additionally,  $b_D$  stands for the number of bends in drawing  $D$ ,  $\lambda(D)$  is the number of rows in a vertex-by-vertex compaction of  $D$ , and  $\bar{\lambda}(D)$  is the number of rows in a row-by-row compaction.  $\chi(G)$  represents the chromatic number of graph  $G$ .

## 3 Hardness of Row-by-Row Compaction

As a warm-up, we start with a simplified path compaction problem in which every pair of objects on the same row of the drawing must move in tandem. Our proof constructs a drawing of a path graph such that the valid row assignments for our drawing are the same as the valid colorings of a given graph  $G$ .

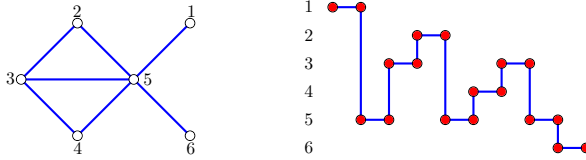
**Lemma 2.** *Given a graph  $G$  we can construct in polynomial time a rectilinear drawing  $D$  of a path graph with  $O(m_G)$  vertices, such that  $\bar{\lambda}(D) = \chi(G)$ .*

**Proof.** Find a Chinese postman walk for  $G$ ; that is, a walk that starts at an arbitrary vertex and visits each edge at least once, allowing vertices to be visited multiple times. Such a walk may be found, for instance, by doubling each edge of  $G$  and constructing an Euler tour of the doubled graph. Let  $u_i v_i$  be the  $i$ th edge in the walk, where  $v_i = u_{i+1}$ , and let  $k \leq 2m_G$  be the number of edges in the walk. Additionally, choose arbitrary distinct integer numbers for the vertices of  $G$  with  $\ell(v)$  being the number for the vertex  $v$ .

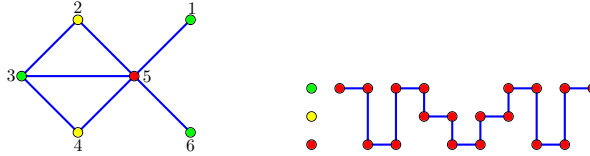
To construct the drawing  $D$ , for  $i$  from 0 to  $k$ , place vertices in the plane at the points  $(i, \ell(u_i))$  and  $(i + 1, \ell(u_i))$ , connected by a unit-length horizontal edge. Additionally, for  $i$  from 0 to  $k - 1$  draw a vertical edge from  $(i + 1, \ell(u_i))$  to  $(i + 1, \ell(v_i))$ . See Figure 2 for an example of such a construction.

Two rows in the drawing conflict if and only if the corresponding vertices in  $G$  are adjacent. For every coloring of  $G$ , we may compact  $D$  by using one row for the vertices of each color, and conversely for every row-by-row compaction of  $D$  we may color  $G$  by using one color class for each row of the compaction (Figure 3). Therefore,  $\bar{\lambda}(D) = \chi(G)$ . Also,  $n_D = 2k + 2 = O(m_G)$ . □

The same drawing  $D$  can equivalently be viewed as an orthogonal drawing of the two-vertex graph  $K_2$  with  $O(m_G)$  bends. In the restricted model of compaction used in this section, horizontal compaction is disallowed, so optimizing the area of a compaction of  $D$  is the same as optimizing its number of rows.



**Fig. 2.** Path constructed from a graph  $G$  using the walk 1, 5, 3, 2, 5, 4, 3, 5, 6



**Fig. 3.** The rows of a compacted drawing correspond to the colors in a coloring of  $G$

**Theorem 1.** *Let  $\epsilon > 0$  be any positive fixed constant, and suppose that  $P \neq NP$ . Then there does not exist a polynomial time algorithm that approximates the number of layers or the area in an optimal row-by-row compaction of a given orthogonal or rectilinear drawing  $D$  to within a factor of  $(n_D + b_D)^{1/2-\epsilon}$ .*

**Proof.** Suppose for a contradiction that algorithm  $\mathcal{A}$  can solve the row-by-row compaction problem to within a factor  $\rho \leq (n_D + b_D)^{1/2-\epsilon}$  of optimal. Let  $\mathcal{A}'$  be an algorithm for coloring an input graph  $G$  by performing the following steps:

1. Use Lemma 2 to construct a path drawing  $D$  from the given graph  $G$ .
2. Use algorithm  $\mathcal{A}$  to compact  $D$ .
3. Color  $G$  using one color for each row of the compacted drawing.

Then the approximation ratio of algorithm  $\mathcal{A}'$  for coloring is the same number  $\rho$  as the approximation ratio of algorithm  $\mathcal{A}$  for compaction, whether measured by area or by number of rows. However,

$$\rho \leq (n_D + b_D)^{1/2-\epsilon} = O(m_G^{1/2-\epsilon}) = O(n_G^{1-2\epsilon}),$$

an approximation ratio that contradicts Lemma 1. □

The same reduction, together with the NP-completeness of graph 3-colorability, shows that it is NP-complete to determine whether a given drawing  $D$  has a row-by-row compaction that uses at most three rows; we omit the details.

## 4 Hardness of Vertex-by-Vertex Compaction

Our hardness result for vertex-by-vertex vertical compaction follows roughly the same outline as Theorem 1: translate graph vertices into drawing features such that two features can be compacted onto the same row if and only if the corresponding graph vertices can be assigned the same color. However, direct overlaps between pairs of features would only let us represent interval graphs, which are easily colored, so instead we use an *edge gadget* depicted in Figure 4

to represent an edge between two vertices of the input graph. This gadget has six vertices and six line segments; the two vertices  $A$  and  $B$  of the gadget may be placed on two line segments representing vertices of the input graph. This connection forces the two line segments containing  $A$  and  $B$  to be placed on different rows of any compacted drawing, even if these two line segments have no vertical overlap with each other: one of the two line segments must be above the central rectangle of the gadget, and the other must be below the central rectangle, although either of these two orientations is possible.

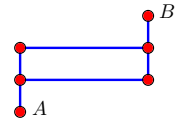


Fig. 4. Edge gadget

The use of these edge gadgets leads to a second difficulty in our reduction: the number of rows in the compacted drawing will depend both on the features coming from input graph vertices and the rows needed by the edge gadgets themselves. In order to make the first of these two terms dominate the total, we represent an input graph vertex by a *bundle* of  $\theta$  parallel line segments, for some integer  $\theta > 0$ . The edge gadgets may be modified to enforce that all segments in one bundle be in different rows from all segments of a second bundle, as shown in Figure 5, while only using a constant number of rows for the gadget itself.

Figure 6 shows the complete reduction, for a graph  $G$  with five vertices and six edges, and for  $\theta = 1$ . Each vertex of  $G$  is represented as a horizontal black line segment (or bundle of segments, for  $\theta > 1$ ), and each edge of  $G$  is represented by an edge gadget. The vertices of  $G$  are numbered arbitrarily from 1 to  $n_G$ , and these numbers are used to assign vertical positions to the corresponding bundles of segments in the drawing. The edge gadgets are given  $x$ -coordinates that allow them to attach to the two vertex bundles they should be attached to, and  $y$ -coordinates that place them between these two vertex bundles.

**Lemma 3.** *Given a graph  $G$  and a parameter  $\theta$  we can construct in polynomial time an orthogonal drawing  $D$  such that the vertices of  $D$  have maximum degree 3,  $n_D = O(n_G^2\theta)$ , and*

$$\chi(G)\theta \leq \lambda(D) \leq \chi(G) + O(n_G)^2.$$

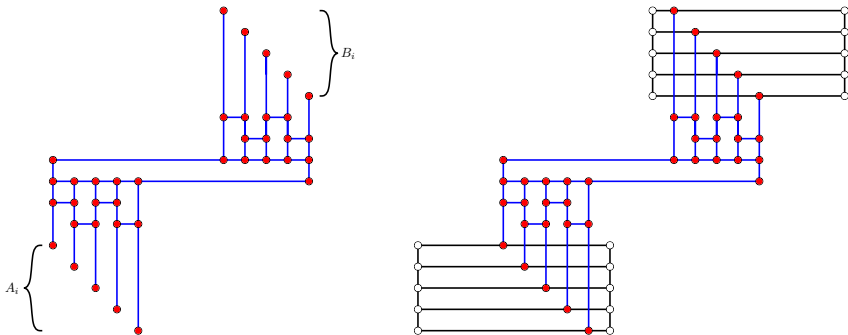


Fig. 5. The full edge gadget for  $\theta = 5$ .

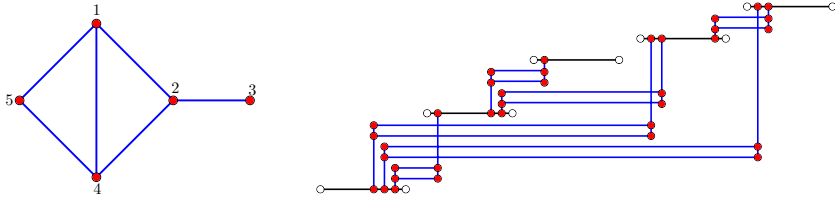


Fig. 6. Example of the complete reduction for  $\theta = 1$

**Proof.** The construction of  $D$  is as described above. It is straightforward to verify the bounds on  $n_D$  and on the degree. If  $G$  has a coloring with  $\chi$  colors, it is possible to assign the vertex bundles of  $D$  to  $\chi$  sets of  $\theta$  rows each, according to those colors, with an additional  $O(n_G)$  rows between any two such sets to allow room for the edge gadgets to be placed without interference with each other. Therefore,  $\lambda(D) \leq \chi(G) + O(n_G)^2$ .

If  $D'$  is a compacted drawing of  $D$ , acyclically orient the edges of  $G$  from the vertex whose bundle is below the edge gadget to the vertex whose bundle is above the edge gadget, and assign each vertex  $v$  in  $G$  a color indexed by the length of the longest path from a source to  $v$  in this acyclic orientation. Then the number of colors needed equals the number of vertices in the longest path, and the number of rows in  $D'$  needed just for the vertices in this path is  $\theta$  times the number of vertices of  $G$  in the path. Therefore,  $\chi(G)\theta \leq \lambda(D)$ .  $\square$

**Theorem 2.** *If  $P \neq NP$ , then no polynomial time algorithm approximates the number of layers or the area in an optimal vertex-by-vertex vertical compaction of a given orthogonal graph drawing to within a factor of  $(n_D + n_B)^{1/4-\epsilon}$ .*

**Proof.** If an algorithm could achieve this approximation ratio for compaction, we could get an  $O(n^{1-4\epsilon})$  ratio for coloring by applying Lemma 3 with  $\theta = n_G^2$ , compacting the resulting drawing, and using the coloring derived from the compaction in the proof of Lemma 3. But this would contradict Lemma 1.  $\square$

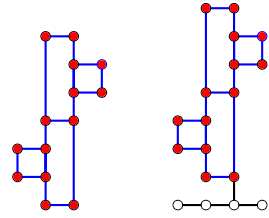
## 5 Hardness of Vertex-by-Vertex Free Compaction

In the reduction from the previous section, allowing the vertices to move horizontally as well as vertically does not make any difference in how much vertical compaction is possible. However, if we want to prove inapproximability for minimal-area compaction, we also need to worry about horizontal compaction. By making the width incompressible we may make the vertical compaction factor the same as the area compaction factor.

**Lemma 4.** *From a drawing  $D$  a drawing  $D'$  can be constructed by adding at most  $O(n_D)$  vertices, such that  $\lambda(D') = \lambda(D) + 1$  and  $D'$  is incompressible in the horizontal direction. If  $D$  has maximum degree three, then so does  $D'$ .*



**Proof.** Place a line of vertices on a new row below  $D$ ; for each set of vertices with a given  $x$ -coordinate in  $D$ , add a vertex on the new row at the same  $x$ -coordinate. Connect the added vertices with horizontal edges, and add a vertical edge to connect these vertices to  $D$  at the point of  $D$  that is rightmost on its bottom row, as shown in Figure 7. This added layer conflicts with all existing horizontal layers, and forces  $D'$  to be incompressible in the horizontal direction.  $\square$



**Fig. 7.** Adding a row to  $D$  prevents horizontal compaction

**Theorem 3.** *Unless  $P = NP$ , it is impossible to find vertex-by-vertex free compactions with area within a factor of  $n_D^{1/4-\epsilon}$  of optimal in polynomial time.*

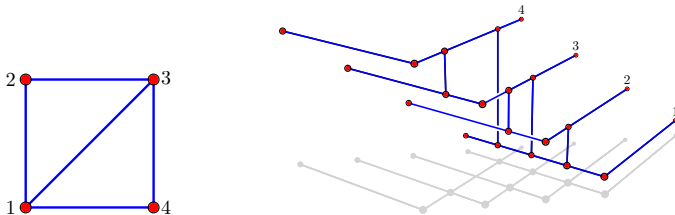
### 6 Hardness of Three-Dimensional Compaction

Our hardness result for three-dimensional compaction follows from the construction of a drawing whose valid two-dimensional layer assignments are the same as the valid colorings of a graph  $G$ . We assign to each vertex in  $G$  a horizontal layer containing an  $L$ -shaped pair of line segments, such that when projected vertically onto a plane every two of these  $L$  shapes cross each other. For each edge in  $G$  we place a vertical edge in the drawing connecting the  $L$  shapes that correspond to the endpoints of the edge. Figure 8 shows an example.

**Lemma 5.** *Given a graph  $G$  we can construct in polynomial time a 3D orthogonal drawing  $D$  with maximum degree three such that  $n_D = 3n_G + 2m_G = O(n_G^2)$ , and such that the number of layers in an optimal  $y$ -compaction is  $\chi(G)$ .*

**Theorem 4.** *If  $P \neq NP$ , then there does not exist a polynomial time algorithm that approximates the number of layers in an optimal layer compaction of a given three dimensional orthogonal drawing to within a factor of  $n_D^{1/2-\epsilon}$ .*

We omit the proofs, which follow the same lines as the previous results.



**Fig. 8.** Reduction from coloring to three-dimensional compaction where  $y$  is the vertical direction

## 7 Approximation Algorithm

In this section we show that several versions of compaction can be approximated to within a ratio of  $O(\sqrt{n})$  of optimal in polynomial time. Our intent in presenting this is not so much to describe a useful compaction algorithm but rather to show that our  $\Omega(n^{1/2-\epsilon})$  inapproximability bounds are nearly tight.

Our approximation method applies to both row-by-row and vertex-by-vertex vertical compaction, in two or three dimensions, with the optimization criterion being minimizing the number of rows or layers. In each case, we may form an *incompatibility graph*, where the vertices of the incompatibility graph represent sets of drawing features that must move in tandem: rows or layers, in row-by-row compaction, or connected components of the subgraph of the drawing formed by horizontal edges, in vertex-by-vertex vertical compaction. Two vertices of the incompatibility graph are connected by an edge when the drawing features they represent cannot be compacted to the same layer of the drawing, that is, when they contain parts of the drawing that are directly above one another.

Our approximation algorithm is, essentially, a standard greedy graph coloring algorithm applied to the incompatibility graph. Specifically, it performs the following steps.

1. Construct the incompatibility graph  $G$  from the given drawing  $D$ .
2. Find a *degeneracy ordering* of  $G$  by initializing an empty list  $L$ , and then repeatedly finding and adding to the end of  $L$  the vertex  $v$  minimizing the number of neighbors of  $v$  that do not already belong to  $L$ .
3. Process the vertices of  $G$  in the reverse of the ordering given by  $L$ . For each vertex, in this order, assign it the smallest positive integer that is distinct from the integers assigned to its already-processed neighbors.
4. Use the numbers assigned to the vertices in  $G$  as the coordinates of the corresponding features in a compaction of  $D$ .

To analyze this algorithm, we consider the *degeneracy*  $\delta$  of  $G$  [7]. If we orient  $G$  from earlier vertices to later vertices in the degeneracy ordering described above,  $\delta$  is the maximum outdegree of a vertex in the orientation. Alternatively,  $\delta$  is the smallest number with the property that every set  $S$  of vertices in  $G$  includes a vertex that has at most  $\delta$  neighbors in  $S$ . Let  $\kappa$  denote the largest number of features of  $D$  that can be pierced by a vertical line through a vertex or bend. As we now show,  $\delta \leq \sqrt{2(n_D + b_D)\kappa}$ . For, if  $|S| \leq \sqrt{(n_D + b_D)\kappa}$ , then clearly all vertices have at most  $\sqrt{2(n_D + b_D)\kappa}$  neighbors in  $S$ . And, if  $|S| \geq \sqrt{2(n_D + b_D)\kappa}$ , then there are at most  $(n_D + b_D)\kappa$  edges in  $G$  (each vertex or bend of  $D$  contributes at most  $k$  incompatibilities) so by an averaging argument there is a vertex in  $S$  with degree at most  $2(n_D + b_D)\kappa/|S| \leq \delta$ .

**Theorem 5.** *For 2d or 3d row-by-row or vertex-by-vertex vertical compaction, the algorithm described above computes a valid compaction whose number of rows or layers is within an  $O(\sqrt{n_D + b_D})$  factor of optimal.*

**Proof.** No two features can overlap in the compacted drawing: for, if two features do not overlap vertically in  $D$ , they cannot overlap no matter how they are

compacted, and if two features do overlap vertically then the corresponding nodes in  $G$  will be adjacent and will be assigned distinct coordinate values. Therefore, the result of the algorithm is a valid compaction.

Any valid compaction must have at least  $\kappa$  layers. But as we have seen, each vertex in  $G$  has  $O(\sqrt{(n_D + b_D)\kappa})$  earlier neighbors in the order and each of these neighbors can only eliminate one choice from the set of possible coordinate values, so its coordinate value in the compaction is  $O(\sqrt{(n_D + b_D)\kappa})$ . Therefore, the approximation ratio is  $O((\sqrt{(n_D + b_D)\kappa})/\kappa) = O(\sqrt{n_D + b_D})$ .  $\square$

## 8 Fixed-Parameter Tractability of Vertex-by-Vertex Vertical Compaction

**Lemma 6.** *Given an orthogonal drawing  $D$  we can compact  $D$  into  $k$  layers in  $O(k!(b + n))$  time, if such a compaction is possible.*

**Proof.** We construct local assignments of the features into  $k$  rows via a left-to-right plane sweep. The drawing may be assumed to be in a  $n \times n$  grid, so the features can be sorted in linear time. While sweeping the plane we maintain a set of those features intersecting the sweep line along with a record of valid assignments of these features into the  $k$  rows.

When a feature first intersects the sweep line we try to place it into the collection of valid assignments. If there are  $\ell$  features intersecting the sweep line prior to the insertion, we have at most  $\ell! \binom{k}{\ell}$  valid assignments to consider. In each of these valid assignments there are  $k - \ell$  free rows. Altogether at most  $k!$  configurations will be considered for each feature insertion. When the sweep line moves past a feature its row is freed for future use.

If at any point we cannot find any valid assignment for a new feature, we conclude that a compaction into  $k$  rows is not possible. On the other hand if the last feature can be placed into a valid assignment, then a compaction into  $k$  layers is possible. To recover the global assignment of horizontal features into rows, we may backtrack through the sets of local assignments.  $\square$

**Theorem 6.** *An optimal vertex-by-vertex vertical compaction of an orthogonal drawing  $D$  can be found in  $O(\lambda!(b + n))$  time where  $\lambda = \lambda(D)$ .*

**Proof.** Apply Lemma 6 for  $k = 1, 2, 3, \dots$  until finding a value of  $k$  for which a valid layering exists.  $\square$

## 9 Conclusions

Our investigations have determined upper and lower bounds for several different approximation and fixed-parameter versions of the compaction problem. In some cases, our bounds are tight: we have upper and lower bounds on the approximation ratio with the same exponent. In some other cases, there remain gaps,

the most important of which is in the problem with the greatest relevance for practical graph drawing: vertex-by-vertex free compaction of two-dimensional orthogonal drawings to minimize area. For this problem, we have an  $\Omega(n^{1/4-\epsilon})$  lower bound on the approximation ratio, and no upper bound. Can our  $O(\sqrt{n})$  approximation algorithms be extended to cover this case? Can the exponent in the lower bound be improved? We leave these questions open for future research.

**Acknowledgements.** This work was supported in part by NSF grant 0830403 and by the Office of Naval Research under grant N00014-08-1-1015.

## References

1. Biedl, T.C., Madden, B.P., Tollis, I.G.: The three-phase method: a unified approach to orthogonal graph drawing. *Int. J. Comput. Geom. Appl.* 10(6), 553–580 (2000), doi:10.1142/S0218195900000310
2. Bridgeman, S.S., Di Battista, G., Didimo, W., Liotta, G., Tamassia, R., Vismara, L.: Turn-regularity and optimal area drawings of orthogonal representations. *Computational Geometry: Theory and Applications* 16(1), 53–93 (2000), doi:10.1016/S0925-7721(99)00054-1
3. Eades, P., Hong, S.-H., Poon, S.-H.: On Rectilinear Drawing of Graphs. In: Eppstein, D., Gansner, E.R. (eds.) *GD 2009*. LNCS, vol. 5849, pp. 232–243. Springer, Heidelberg (2010), doi:10.1007/978-3-642-11805-0\_23
4. Eiglsperger, M., Fekete, S.P., Klau, G.W.: Orthogonal Graph Drawing. In: Kaufmann, M., Wagner, D. (eds.) *Drawing Graphs*. LNCS, vol. 2025, pp. 121–171. Springer, Heidelberg (2001), doi:10.1007/3-540-44969-8\_6
5. Eppstein, D.: The Topology of Bendless Three-Dimensional Orthogonal Graph Drawing. In: Tollis, I.G., Patrignani, M. (eds.) *GD 2008*. LNCS, vol. 5417, pp. 78–89. Springer, Heidelberg (2009), doi:10.1007/978-3-642-00219-9\_9
6. Huang, W., Hong, S.-H., Eades, P.: Effects of crossing angles. In: *IEEE Pacific Visualization Symposium (PacificVIS 2008)*, pp. 41–46 (2008), doi:10.1109/PACIFICVIS.2008.4475457
7. Lick, D.R., White, A.T.:  $k$ -degenerate graphs. *Canadian Journal of Mathematics* 22, 1082–1096 (1970), doi:10.4153/CJM-1970-125-1
8. Mañuch, J., Patterson, M., Poon, S.-H., Thachuk, C.: Complexity of Finding Non-Planar Rectilinear Drawings of Graphs. In: Brandes, U., Cornelsen, S. (eds.) *GD 2010*. LNCS, vol. 6502, pp. 305–316. Springer, Heidelberg (2011), doi:10.1007/978-3-642-18469-7\_28
9. Patrignani, M.: On the complexity of orthogonal compaction. *Computational Geometry* 19(1), 47–67 (2001), doi:10.1016/S0925-7721(01)00010-4
10. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* 16(3), 421–444 (1987), doi:10.1137/0216030
11. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing* 3(1), 103–128 (2007), doi:10.4086/toc.2007.v003a006