# Minimizing the Area for Planar Straight-Line Grid Drawings

Marcus Krug and Dorothea Wagner

Universität Karlsruhe
{krug,wagner}@ira.uka.de

**Abstract.** Straight-line grid drawings of bounded size is a classical topic in graph drawing. The Graph Drawing Challenge 2006 dealt with minimizing the area of planar straight-line grid drawings. In this paper, we show that it is NP-complete to decide if a planar graph has a planar straight-line drawing on a grid of given size. Furthermore, we present a new iterative approach to compactify planar straight-line grid drawings. In an experimental study, we evaluate the quality of the compactified drawings with respect to the size of the area as well as to other measures.

## 1 Introduction

A planar straight-line grid drawing of a planar graph is an assignment of the nodes of the graph to integral points in the plane, such that the edges of the graph are mapped to non-crossing straight-lines. The size of such a drawing is typically measured by the number of grid points contained in the bounding box of the drawing, i.e. the smallest axis-parallel rectangle covering the drawing. Minimizing the size of planar straight-line grid drawings has a long tradition. De Fraysseix et al. [1] and Schnyder [2] were the first, who independently showed that every (3-connected) plane graph with $n$ nodes has a straight-line grid drawing on a grid of size $(2n-4) \times (n-2)$ and $(n-2) \times (n-2)$, respectively. Zhang and He [3] showed, that every plane graph with $n$ nodes has a straight-line grid drawing on a grid of size $(n-\Delta_0-1) \times (n-\Delta_0-1)$, where $0 \leq \Delta_0 \leq \lfloor (n-2)/2 \rfloor$ is the number of cyclic faces of the graph with respect to its minimum Schnyder realizer. Minimizing the area has been shown to be NP-complete in several orthogonal drawing models, e.g. by Kramer and van Leeuwen in [4]. However, it has remained open if NP-completeness applies to minimizing the area for straight-line grid drawings as well.

In this paper, we close this gap and show that it is NP-complete to decide if a planar graph has a planar straight-line drawing on a grid of a given size. In order to come up with compact straight-line grid drawings nevertheless, we propose a novel iterative compaction algorithm. The algorithm starts with a planar straight-line grid drawing and computes a more compact drawing by iteratively evaluating new positions for each node. Moreover, we evaluate the algorithm

based on a sample of 400 graphs, which were generated using a common planar graph generator. We will show that the proposed algorithm is capable of producing substantially compactified drawings with additional nice properties.

## 2  NP-Completeness

Let $G$ be a planar graph. We will consider the problem of finding a planar straight-line grid drawing $\mathcal{E}$ for $G$ with minimal size, where the size of the drawing is measured by the number of grid points inside the bounding box $\mathcal{B}(\mathcal{E})$, denoted by $|\mathcal{B}(\mathcal{E})|$. We will refer to this problem as Minimum Area Planar Straight-Line Grid Drawing and we will consider the following related decision problem:

Minimum Area Planar Straight-Line Grid Drawing

INSTANCE: A planar graph $G$ and an integer $A$.
QUESTION: Is there a planar straight-line grid drawing $\mathcal{E}$ for $G$, such that $|\mathcal{B}(\mathcal{E})| \leq A$?

**Theorem 1.** Minimum Area Planar Straight-Line Grid Drawing *is NP-complete.*

*Proof.* We will only outline the main idea of the proof here, a full proof can be found in [5]. It is obvious, that Minimum Area Planar Straight-Line Grid Drawing is in NP. We will prove NP-hardness by reduction from 3-Partition, which is defined as follows:

3-Partition

INSTANCE: $3m$ positive integers $A = \{a_1, \ldots, a_{3m}\}$, a positive integer bound $b$, such that $b/4 < a_i < b/2$ for all $i = 1, \ldots, 3m$
QUESTION: Can $A$ be partitioned into $m$ disjoint sets $A_1, \ldots, A_m$, such that $\sum_{a_i \in A_j} a_i = b$ for all $j = 1, \ldots, m$?

In [6] Garey and Johnson proved that 3-Partition is NP-complete. It is essential for the reduction that 3-Partition is NP-complete in the strong sense, i.e. it remains NP-complete, even if $b$ is bounded by a polynomial in $m$. We will assume that $b \geq 8$ since the problem is trivial for smaller values of $b$.

For an instance $\mathcal{I} = (A, m, b)$ of 3-Partition we will construct an instance $\mathcal{I}' = (G_{\mathcal{I}}, A_{\mathcal{I}})$ of Minimum Area Planar Straight-Line Grid Drawing, such that $A_{\mathcal{I}} = 3p$, where $p$ is the smallest prime greater or equal to $l + 17$ with $l = (m+1)(2b+2) - m$. By the Bertrand-Chebyshev theorem such a prime number does exist in the interval $[l + 17, 2(l + 17)]$. It is vital for the proof, that the only grid of size $A_{\mathcal{I}}$ is a grid of width $p$ and height three, or vice versa. Each number $a_i$ will be represented as a path of length $2a_i$, and the graph $G_{\mathcal{I}}$ is given by the union of the paths representing the numbers in $A$ and the frame graph $F_{m,b,p}$ associated with the instance $\mathcal{I}$. A drawing of the frame graph on a grid of size $p \times 3$ is illustrated in Fig. 1. It consists of a linked sequence of $m + 2$

diamond-shaped graphs $D_1, \ldots, D_{m+2}$, which are additionally belted by a path. Each $D_i$ $(i < m+2)$ consists of two adjacent nodes with degree $2b + 3$ sharing $2b + 2$ common neighbors. These common neighbors are linked as two disjoint paths of length $b + 1$ each. The graph $D_{m+2}$ is constructed analogously with $l - p + 1$ common neighbors. Up to rotation and reflection, there is only one way of drawing the frame graph on a grid of size $A_\mathcal{I}$, assuming that none of the grid points may be covered by an edge, i.e. as illustrated in Fig. 1. Since the number of nodes of $G_\mathcal{I}$ equals $A_\mathcal{I}$, this condition holds for any drawing of $G_\mathcal{I}$ on any grid of size $A_\mathcal{I}$. The given drawing of the frame graph leaves exactly $m$ boxes
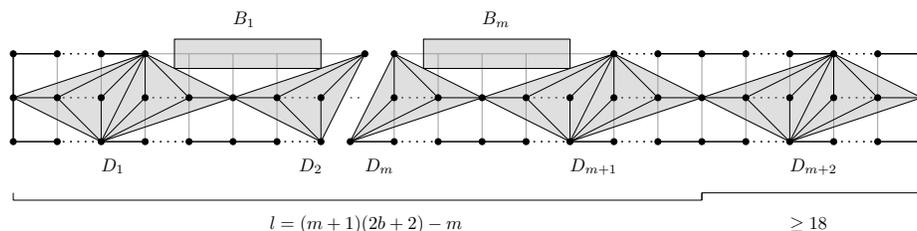


**Fig. 1.** The frame graph used in the reduction from 3-PARTITION: The diamond-shaped graphs $D_i$ are accentuated in grey.

$B_1, \ldots, B_m$ of width $2b$ and height one unoccupied, such that $B_i \cap B_j = \emptyset$ for $i \neq j$. Therefore, we obtain a one-to-one correspondence between the problem of deciding, if the set $A$ can be partitioned according to 3-PARTITION and the problem of deciding, if the paths representing the numbers in $A$ can be drawn into the boxes $B_i$, i.e. if $G_\mathcal{I}$ has a straight line drawing on a grid of size $A_\mathcal{I}$, which concludes the proof. $\qquad\square$

Note, that the proof also yields that it is NP-complete to decide if a graph has a straight-line drawing on a grid with given width and height, and that the problem remains NP-complete if restricted to plane graphs, i.e. graphs with a given combinatorial embedding.

## 3  Compaction Algorithm

In this section, we will present an iterative algorithm for the compaction of planar straight-line grid drawings. The algorithm takes a plane graph as an input and produces a more compact drawing. It works in iterations and rounds. The number of iterations is a parameter of the algorithm. At the beginning of each iteration, we compute the bounding box of the current drawing and randomly select a grid point $\varrho$ inside the bounding box, which we refer to as the reference point for this iteration. Each iteration then consists of a certain number of rounds. In each round, the nodes of $G$ are examined with ascending distance to the reference point $\varrho$. For each node $v$, the algorithm considers a

set $P_v$ of possible new positions for $v$ according to an evaluation function $\varphi_{v,\varrho}$ on $P_v$. Both $P_v$ and $\varphi_{v,\varrho}$ will be explained below. For each grid point $p$ in $P_v$, the algorithm computes $\varphi_{v,\varrho}(p)$ and tentatively assigns $v$ to the grid point with the smallest negative value. If the resulting drawing is planar $v$ will be permanently assigned to this grid point for the rest of this round. Otherwise the grid point will be discarded and the next best grid point with respect to $\varphi_{v,\varrho}$ will be considered. If there is no grid point $p$ in $P_v$ such that $\varphi_{v,\varrho}(p) < 0$, the next node will be considered and no action will be taken for $v$. If none of the nodes could be assigned to a new grid point in one round, the current iteration ends. In our implementation we considered the set $P_v$ of unoccupied grid points $p$ with distance less than or equal to $D = \sqrt{2}$ to the current position of $v$. Note, that a different choice of $P_v$ is also possible. The evaluation function is given by

$$\varphi_{v,\varrho}(p) = \psi_{v,\varrho}(p) - \psi_{v,\varrho}(v), \tag{1}$$

where

$$\psi_{v,\varrho}(p) = (p - \varrho)^2 + \sum_{\{v,w\}\in E} (p - w)^2 . \tag{2}$$

The intuition behind the evaluation function is to group the nodes around the reference point thereby compactifying the drawing while simultaneously enforcing short edges in order to prevent the algorithm from running into a local optimum. By randomly choosing reference points in each iteration, we further seek to make the algorithm more robust against local minima.

Since the number of rounds performed during one iteration is not bounded, we must ensure, that the algorithm terminates. To this end, we will consider a global evaluation function $\mathcal{F}_\varrho$ for a given drawing and a given reference point $\varrho$. We will define $\mathcal{F}_\varrho$ as

$$\mathcal{F}_\varrho(\mathcal{E}) = \sum_{v\in V} (v - \varrho)^2 + \sum_{\{v,w\}\in E} (v - w)^2 . \tag{3}$$

Suppose that the new drawing $\mathcal{E}'$ is obtained from the drawing $\mathcal{E}$ by moving the node $v$ from $p$ to $p'$, such that $\varphi_{v,\varrho}(p') < 0$ as imposed by the algorithm. Then,

$$\mathcal{F}_\varrho(\mathcal{E}') - \mathcal{F}_\varrho(\mathcal{E}) = \psi_{v,\varrho}(p') - \psi_{v,\varrho}(p) = \varphi_{v,\varrho}(p') < 0 . \tag{4}$$

Hence, moving a node can only decrease $\mathcal{F}_\varrho$ and $\mathcal{F}_\varrho \geq 0$ by definition.

## 4 Experimental Evaluation

We have performed an extensive experimental study based on more than 5,000 graphs generated using a variety of different planar graph generators. However, due to limitations of space, we will only present a small fraction of our results here. We will restrict ourselves to the sample of graphs generated using the LEDA `random_planar_graph` function. We generated graphs with $n$ nodes where $n \in \{250, 500, 750, 1000\}$. For each $n$ we generated 20 graphs with $m$ edges each,

where $m \in \{n, \frac{3}{2}n, 2n, \frac{5}{2}n, 3n-6\}$, resulting in a total of 400 graphs. The graphs were initially drawn on a grid of size $(n-2) \times (n-2)$ using the algorithm proposed by Schnyder in [2]. The performance of the algorithm is measured by the *compaction factor* $\gamma$, which relates the area $A_0$ of the initial drawing to the area $A$ of the final drawing, i.e. $\gamma = \frac{A_0}{A}$. A high compaction factor indicates, that the algorithm substantially compactified the initial drawing. We performed 20 iterations of the compaction algorithm on all initial drawings.

The observed compaction factors range from 1.5 up to 135. For more than 50% of the graphs the compaction factors achieved exceed a value of 10, i.e. the area of the compactified drawings occupies merely less than $\frac{1}{10}$-th of the initially required area. However, 4% of the compactified drawings still occupy more than half the area of the initial drawing.

Expectedly, the observed compaction factors depend on both the number of nodes and the density of the graph. For a given number of nodes, a higher density implies less degrees of freedom for the placement of nodes. Similarly, depending on the structure of the graph, a larger number of nodes may imply longer edges for a given density and, thus, less freedom for the placement of nodes, e.g. the graph consisting of a nested sequence of $k$ triangles. The lower and upper quartiles as well as the median of the observed compaction factors decrease as the number of nodes increases: More than 75% of the drawings of the graphs with 250 nodes achieved compaction factors larger than 10, as opposed to only 21% of the graphs with 1000 nodes. Similarly, the compaction factors decrease as the density of the graphs increases: More than 80% of the drawings of the very sparse graphs ($m = n$) and still more than 75% of those of the sparse graphs ($m = 1.5n$) achieved compaction factors of 10 and larger, as opposed to 42% for graphs with high density ($m = 2.5n$) and only 26% for the graphs with maximum density ($m = 3n - 6$).

The compactified drawings are satisfactory with respect to commonly used measures for graph drawings (Fig. 2): The aspect-ratios of the compactified drawings, i.e. the ratio of longer vs. shorter side of the bounding box, range between 1 and 4. More precisely, in 95% of the drawings the aspect ratio does not exceed a value of 2. In more than 65% of the compactified drawings the
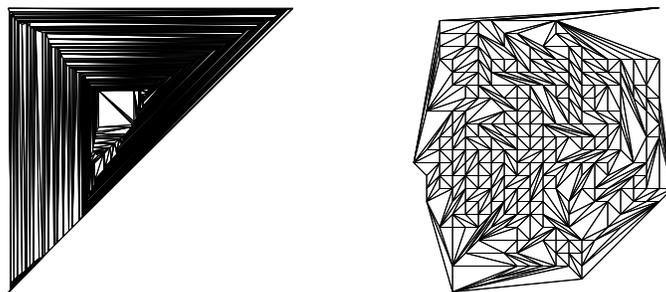


**Fig. 2.** Sample output of the compaction algorithm: Initial drawing (left) and scaled (by a factor of $\approx 11$) compactified drawing (right); $\gamma \approx 128$.

angular resolution, i.e. the smallest angle between incident edges, is not worse than in the initial drawings. As expected, the edge-length resolution, i.e. the ratio of longest vs. shortest edge-length is also improved by the compactified drawings.

The running time of the algorithm is substantially influenced by the number of rounds performed per iteration. The number of rounds performed during the 20 iterations of the algorithm seems to fit the model $cn^k$ with $k > 1$. Using non-linear least squares fitting, we obtain $c = 0.37 \pm 10^{-4}$ and $k = 1.4 \pm 10^{-4}$, i.e. the number of rounds per iteration seems to be slightly super-linear in the number of nodes. However, this number tends to decrease with each iteration, since it depends on the size of the drawing.

## 5  Conclusion and Outlook

We proved that it is NP-complete to decide if a given planar graph has a planar straight-line grid drawing with a given size. We further proposed an iterative algorithm for the compactification of planar straight-line grid drawings. Our experimental evaluation of the algorithm reveals, that the performance of the algorithm depends on both the number of nodes and the density of the graphs. Due to the local nature of the algorithm, the initial drawing has a great influence on the performance. In addition to that the tests for intersections which must be performed after each update of a node may be costly since they may have to be performed for a large number of edges. In order to speed up the computation, the number of edges which have to be tested for intersections can be computed using a logical indexing approach which uses the combinatorial embedding of the graph. We were able to show, that the algorithm performs well on a large number of graphs, i.e. more than 50% of the compactified drawings require less than $\frac{1}{10}$-th of the space required for the initial drawing, starting from an already compact drawing.

## References

1. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica 10 **1** (1990) 41–51
2. Schnyder, W.: Embedding planar graphs on the grid. In: SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (1990) 138–148
3. Zhang, H., He, X.: Canonical ordering tree and its applications in graph drawing (2003)
4. Kramer, M., van Leeuwen, J.: The NP-completeness of finding minimum area layouts for VLSI-circuits (to appear). Technical Report RUU-CS-82-06, Institute of Information and Computing Sciences, Utrecht University (1982)
5. Krug, M.: Minimizing the area for planar straight-line grid drawings. Master's thesis, Universität Karlsruhe (2007)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990)