Experiments on Area Compaction Algorithms for Orthogonal Drawings *

Carla Binucci †   *Walter Didimo* †

binucci@diei.unipg.it   didimo@diei.unipg.it

Abstract

This paper presents an experimental study on several compaction algorithms for orthogonal drawings of graphs. We concentrate on algorithms that minimize the area, and compare them with other representative compaction algorithms in the state of the art. In particular, we propose a new exact algorithm that minimizes the total edge length subject to the optimization of the area, and experimentally prove that it is the best choice in many cases.

1 Introduction

A *planar orthogonal drawing* Γ of a planar graph G is a crossing-free drawing of G such that each vertex is mapped to a point of an integer grid and each edge e is mapped to a sequence of horizontal and vertical segments: a left or a right turn on e is called a *bend*. Figure 1 shows a planar orthogonal drawing with two bends. It is known that G admits a planar orthogonal drawing if and only if it is a *4-planar graph*, that is, the number of edges incident on any vertex of G is at most four. An *orthogonal representation* H of G is an equivalence class of planar orthogonal drawings of G such that: (i) For each edge (u, v) of G all the drawings of the class have the same sequence of left and right turns along (u, v) while moving from u to v; (ii) For each vertex v of G and for each pair $\{e_1, e_2\}$ of clockwise consecutive edges incident on v, all the drawings of the class determine the same angle between e_1 and e_2.

We say that a drawing Γ that belongs to H is a planar orthogonal drawing of G that *preserves* H.

A popular and effective technique for computing a planar orthogonal drawing of a 4-planar graph G is the so called *topology-shape-metrics approach*, introduced by Tamassia in [8]. According to this approach, the drawing is computed in three consecutive steps. In the first step a planar embedding for G is determined. In the second step an orthogonal representation H of G is computed within the planar embedding found in the previous step. In the third step a final drawing of G that preserves H is computed, by assigning integer coordinates to vertices and bends.

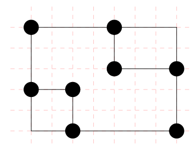


Figure 1: An orthogonal drawing with two bends.

During the third step of the topology-shape-metrics approach, there are some important aesthetic requirements that are usually taken into account in order to improve the quality of the drawing. Two of the most important are the area occupied by the drawing and the total length of the edges; both of them should be kept as small as possible. For this reason, the third step is usually called the *compaction step*. Unfortunately, it is not always possible to minimize both the area and the total edge length of an orthogonal drawing at the same time, while preserving a given orthogonal representation H, even if H has only rectangular faces. Also, compacting an orthogonal representation in such a way that it has either a minimum area or a minimum total edge length is an NP-hard problem [7].

There are several algorithms proposed in the literature for the compaction step of the topology-shape-metrics approach. Klau and Mutzel [6] proposed the first exact (exponential) compaction algorithm to minimize the total edge length, which is based on a ILP formulation of the problem. In [5], Klau et al. extended the algorithm to orthogonal

*This work is partially supported by the MIUR Project ALGO-NEXT: Algorithms for the Next Generation Internet and Web: Methodologies, Design and Applications

†Dip. di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia, Italy

representations with vertex labels. Recently, several exact (exponential) compaction algorithms for orthogonal representations with both vertex and edge labels have been proposed in [1]. Each of these algorithms is devoted to the minimization of one of the following aesthetic requirements: total edge length, width, height, or area. In particular, all these algorithms can also be used for unlabeled graphs; in this case, the model adopted for total edge length minimization reduces to the ILP model of Klau and Mutzel, and the algorithm that minimizes the area provides the first exact algorithm known in the literature for this problem.

Several experimental works have also been presented in the literature to compare the performances of orthogonal compaction algorithms, both in terms of running time and in terms of drawing quality. In particular, Klau and Mutzel [4] presented an extensive experimental study that compares the effectiveness of different orthogonal compaction heuristics in terms of total edge length, evaluating the gap with respect to the optimum solutions computed with their exact algorithm. In [1] experiments are presented focusing on labeled graphs, and only preliminary results are discussed about the compaction problem for unlabeled graphs.

In this paper we concentrate on unlabeled graphs, and compare the exact compaction algorithm for area minimization described in [1] against an exact algorithm for total edge length minimization and against one of the best known compaction heuristics for orthogonal representations [2]. Further a new mixed optimization approach is investigated. From our analysis two main indications arise:

- The choice of the "best" compaction algorithm strongly depends on the density (number of edges over number of vertices) of the graph. In particular, for graphs with high density and number of vertices up to 100, the exact algorithms are preferred to the heuristic in many cases (also considering the running time).

- A mixed optimization algorithm that computes orthogonal drawings of minimum area and then, with lower priority, of minimum total edge length, provides a very good trade-off in terms of different aesthetic requirements. Also, for medium size graphs, the computation of this mixed algorithm is rather fast.

2 Compaction Algorithms and Test Suite

We compared four distinct algorithms for compacting an orthogonal representation H; they are listed below:

- `H-FLOW`: It is the $O(n^2 \log n)$-time heuristic described in [2], which computes a drawing of H by first decomposing all faces of H into *turn-regular* sub-faces (see [2] for details), and then applies flow techniques for minimizing the area and the total edge length as much as possible. The drawing is further refined by applying a one-dimensional compaction algorithm as post-processing.

- `M-TEL`: It computes a drawing of H having the minimum total edge length. The algorithm runs on CPLEX, according to the ILP model described in [1] for labeled graphs. In our case, since the graphs are unlabeled, this model is substantially equivalent to that given in [6].

- `M-AREA`: It computes a drawing of H having the minimum area. This algorithm, described in [1], iterates over a sub-routine, `Min-Width`, that suitably computes a drawing with minimum width within a given height. `Min-Width` again runs on CPLEX, based on an ILP model that is a variant of that used for `M-TEL`, where the objective function is redefined.

- `M-AREA-TEL`: It is a multi-objective optimization algorithm that computes a drawing of H with minimum area A and with minimum total edge length within A. To implement this algorithm, we slightly modified the objective function of sub-routine `Min-Width` in algorithm `M-AREA`, in such a way that it optimizes the total edge length with a lower priority with respect to the minimum width.

All the algorithms above have been tested on the following test suites of graphs[1].

- `Random-Graphs` It is known that the difficulty of the orthogonal compaction problem is affected by the density of the graph. Indeed, orthogonal representations of low-density graphs usually have a large number of possible compaction solutions, and then they represent hard

[1]The test suite is available on-line at http://www.diei.unipg.it/PAG_PERS/binucci/binucci.htm.

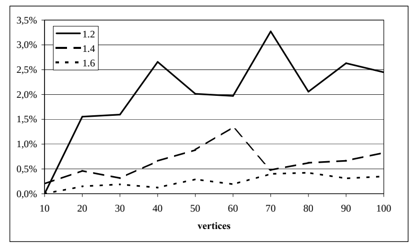
instances for exact algorithms in terms of the required running time. The opposite behavior is typically observed for high-density graph instances. To understand if this behavior is also true for compaction algorithms that minimize the area, we randomly generated 900 (non-planar) graphs having number of vertices in the range $[10, 100]$, density in the set $\{1.2, 1.4, 1.6\}$, and having vertex-degree at most four. The generation was done with a uniform probability distribution, with the procedure in [1].

- `Real-Graphs` To evaluate the performances of the compaction algorithms also on graphs generated from real instances, we extract from the popular test suite introduced in [3] the graphs that have vertex-degree at most four. In this way we collected 647 graphs. Most of these graphs have a density lower than 1.2 and their number of vertices is up to 56.

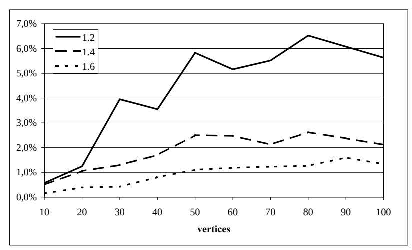
Both the `Random-Graphs` and the `Real-Graphs` are not planar in general. We ran our algorithms on the same orthogonal representations computed by first planarizing the graphs and then applying on them the bend-minimum algorithm described by Tamassia in [8]. The algorithms have been tested on a PC Pentium III, 800MHz, 512MB RAM.

3 Experimental Results

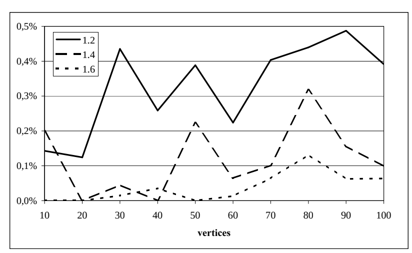
Figure 2 shows the quality of the drawings computed with algorithm `H-FLOW`, `M-AREA`, and `M-AREA-TEL` in terms of total edge length, on the `Random-Graphs`; the optimum value for the total edge length has been computed with `M-TEL`. In particular, we observe how the gap percentages are strongly affected by the graph density, and how the largest gap values are on graphs with low density. Also, we remark how `H-FLOW` is about 2-times better than `M-AREA` to approximate the optimum total edge length value, while `M-AREA-TEL` outperforms the other two, and is very close to the optimum in most cases. Combining these results with those in Figure 3, we can derive that `M-AREA-TEL` is the best choice for a good trade-off between minimum area and minimum total edge length. This observation is reinforced by the results on the running time. Indeed, Figure 4 shows how `M-AREA-TEL` runs rather fast on all instances, and for the low-density graphs it is even faster than `M-AREA`. This is because, the multi-objective function in the subroutine `Min-Width` significantly restricts the number of candidate solutions explored by the algorithm. We also observe that for high-density graphs `H-FLOW` requires more computation time than the exact algorithms (for density 1.6 it takes the same time as `M-AREA-TEL`), while it is much faster on low-density graphs. Of course, for large graphs with some thousand of vertices, `H-FLOW` remains feasible and always outperforms the exact algorithms in terms of running time.



(a)

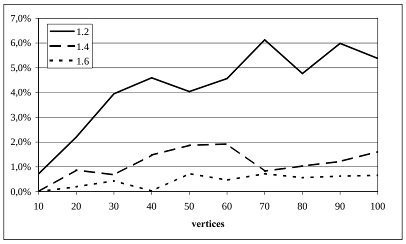


(b)

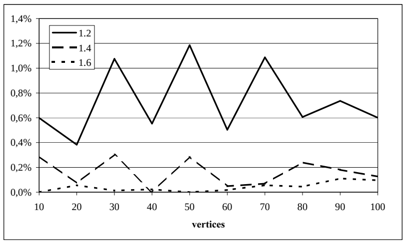


(c)

Figure 2: `Random-Graphs`: Average gap between the minimum total edge length and the total edge length of the drawings computed with (a) H-FLOW, (b) M-AREA, (C) M-AREA-TEL, for the three density values.



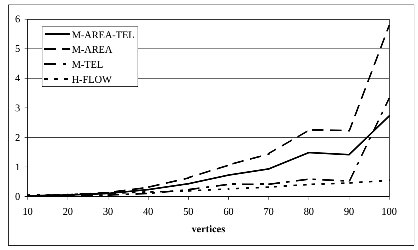
(a)



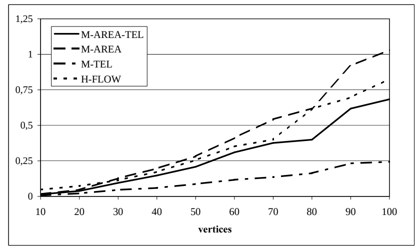
(b)

Figure 3: `Random-Graphs`: Gap between the minimum area and the area of the drawings computed with (a) H-FLOW, (b) M-TEL, for the three density values.

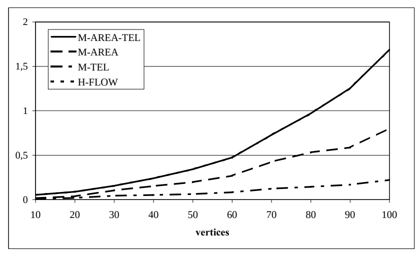
The results on the `Real-Graphs` confirm the behavior of the algorithms obtained for `Random-Graphs`, although the average gaps increase (sometime significantly) due to the fact that the `Real-Graphs` have density lower than 1.2 in most cases (most instances have a tree-like structure). For reasons of space, we omit the charts.



(a)



(b)



(c)

Figure 4: `Random-Graphs`: Average CPU-times of the algorithms in seconds, for graphs with density (a) 1.2, (b) 1.4, (c) 1.6; `M-AREA-TEL`, `H-FLOW` take the same time.

References

[1] C. Binucci, W. Didimo, G. Liotta, and M. Nonato. Orthogonal drawings of graphs with vertex and edge labels. *CGTA*. to appear.

[2] S. Bridgeman, G. D. Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *CGTA*, 16:53–93, 2000.

[3] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *CGTA*, 7:303–326, 1997.

[4] G. Klau, K. Klein, and P. Mutzel. An experimental comparison of orthogonal compaction algorithms. In *GD'00*, volume 1984 of *LNCS*, pages 37–51, 2001.

[5] G. Klau and P. Mutzel. Combining graph labeling and compaction. In *GD'99*, volume 1731 of *LNCS*, pages 27–37, 1999.

[6] G. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In *IPCO'99*, volume 1610 of *LNCS*, pages 304–319, 1999.

[7] M. Patrignani. On the complexity of orthogonal compaction. *CGTA*, 19:47–67, 2001.

[8] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. on Comput.*, 16(3):421–444, 1987.