Minimum Depth Graph Embedding*

Maurizio Pizzonia[1] and Roberto Tamassia[2]

[1] Dipartimento di Informatica e Automazione, Università di Roma Tre, via della Vasca Navale 79, 00146 Roma, Italy. pizzonia@dia.uniroma3.it
[2] Department of Computer Science, Brown University, 115 Waterman Street, Providence, RI 02912–1910, USA. rt@cs.brown.edu

Abstract. The depth of a planar embedding is a measure of the topological nesting of the biconnected components of the graph. Minimizing the depth of planar embeddings has important practical applications to graph drawing. We give a linear time algorithm for computing a minimum depth embedding of a planar graphs whose biconnected components have a prescribed embedding.

1 Introduction

Motivated by graph drawing applications, we study the problem of computing planar embeddings with minimum depth, where the depth of a planar embedding is a measure of the topological nesting of the blocks (biconnected components) of the graph. The main result of this paper is a linear time algorithm for computing minimum-depth embeddings.

In a planar embedding, blocks are inside faces, and faces are inside blocks. The containment relationships between blocks and faces induces a tree rooted at the external face. The depth of the planar embedding is the maximum length of a root-to-leaf path in this tree (see Figure 1).

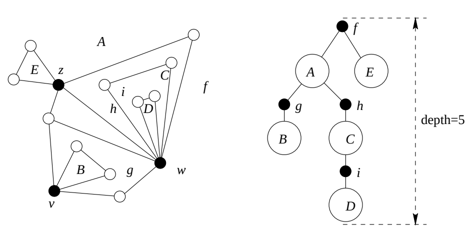


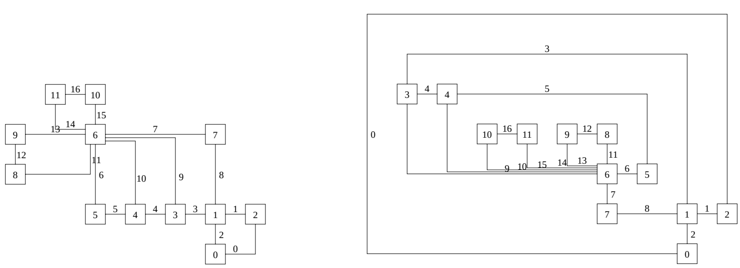
Fig. 1. In this example an embedded graph is shown whose blocks A,B,C,D and E are connected by means of the cutvertices v, w and z. The embedding has cutfaces f,g,h and i. The containment relationship between cutfaces and blocks is represented by a tree of depth 5 rooted at the external face f.

* Research supported in part by the National Science Foundation under grants CCR–9732327 and CDA–9703080, by the U.S. Army Research Office under grant DAAH04–96–1–0013, and by "Progetto Algoritmi per Grandi Insiemi di Dati: Scienza e Ingegneria", MURST Programmi di Ricerca di Rilevante Interesse Nazionale. Work performed in part while Maurizio Pizzonia was visiting Brown University.

M. Paterson (Ed.): ESA 2000, LNCS 1879, pp. 356–367, 2000.

The main motivation for studying depth minimization in planar embeddings comes from the field of graph drawing. A widely used technique for constructing orthogonal drawings of general graphs is the *topology-shape-metric* approach [11,26]. This approach has been extensively investigated both theoretically [19,25,16,21,20,9,10,22] and experimentally [13,4]. Also, its practical applicability has been demonstrated by various system prototypes [12,23] and commercial graph drawing tools [1].

The topology-shape-metrics approach consists of three phases. In the first phase, a planar embedding of the input graph is computed (if the graph is not planar, the graph is planarized using dummy vertices to represent crossings). The successive two phases determine the orthogonal shape (angles) and the coordinates of the drawing, respectively, but do not modify the embedding. Hence, the properties of the embedding computed in the first step are crucial for the the quality of the final layout. In particular, the depth of the embedding negatively affects the area and number of bends of the drawing (see Figure 2). Informally speaking, when a block is nested inside a face, the face must "stretch" to accommodate the block inside it. The importance of optimizing the depth of the embedding in the first phase of the topology-shape-metrics approach was already observed in early papers on the GIOTTO algorithm [3,2].



(a) (b)

Fig. 2. Two drawings of the same graph computed by the GDToolkit system using an algorithm [4] based on the topology-shape-metric approach. Each of the two drawings has bends and area optimized for its embedding. The embedding of drawing (a) has depth 1 while the embedding of drawing (b) has depth 5. Note how the depth of the embedding significantly affects the area, number of bends, and aesthetic quality of the drawing.

Several authors have studied the problem of computing a planar embedding of a graph that is optimal with respect to a certain cost measure. This class of problems is especially challenging because the number of distinct embeddings of a planar graph is exponential in the worst case.

Bienstock and Monma [8] (see also [6,7]) present a general technique to minimize various distance measures over all embeddings of planar graph in polynomial time. They show that minimizing the diameter of the dual of a planar graph, over all planar embeddings, is NP-hard, while it is possible to minimize some distance function to the outer face in polynomial time. Mutzel and Weiskircher [24] present an integer linear programming approach to optimize over all embeddings of biconnected graph a linear cost function on the face cycles.

Computing a planar orthogonal drawing of a planar graph with the minimum number of bends over all possible embeddings is in general NP-hard [17,18]. A polynomial-time algorithm for a restricted class of planar graphs is given in [14]. Heuristic techniques and experimental results are presented in [5,15].

In Section 2 we give basic definitions on embeddings and formally define the concept of depth. In Section 3, we consider a restricted version of the depth minimization problem and present a linear-time algorithm for it. In Section 4 we give a linear time algorithm for the general depth minimization problem, using the algorithm of Section 3 as a building block. Section 5 concludes the paper with open problems.

2 Preliminaries

In this section, we review basic concepts about graphs and embeddings, and give definitions that will be used throughout the paper.

Let G be a connected planar graph. For simplicity, we assume that G has no parallel edges or self-loops. A *cutvertex* of G is a vertex whose deletion disconnects G. Graph G is said to be biconnected if it has no cutvertices. A *block* B of G is a maximal subgraph of G such that B is biconnected. The *block-cutvertex tree* T of G is a tree whose nodes are associated with the blocks and cutvertices of G, and whose edges connect each cutvertex-node to the block-nodes of the blocks containing the cutvertex.

An *embedding* Γ of G is an equivalence class of planar drawings of G with the same circular ordering of edges around each vertex. Two drawings with the same embedding also induce the same circuits of edges bounding corresponding regions in the two drawings. These circuits are called the *faces* of the embedding.

The *dual embedding* Γ' of Γ is the embedded graph induced by the adjacency relations among the faces of Γ through its edges. A *cutface* f of Γ is a face associated with a cutvertex of Γ'. The *block-cutface* tree T^* of Γ is the block-cutvertex tree of Γ'. Since the dual of any biconnected embedding is biconnected T and T^* contains the same set of blocks.

A *planar embedding* is an embedding where a face is chosen as *external face*. We do not consider external faces that are not cutfaces. We consider the block-cutface tree of a planar embedding rooted at the external face.

We now give some definitions about trees. In a tree T the distance between two nodes is the length of the unique path among them. The *eccentricity* of a node v (denoted by $e(v)$) is the maximum among the distances from v to any leaf. The *diameter* of T ($\mathsf{diam}\ T$) is the maximum among the eccentricity of the leaves. For each node v of T, we have $\frac{\mathsf{diam}\ T}{2} \leq e(v) \leq \mathsf{diam}\ T \leq 2e(v)$. The *center* of T is the set of nodes with minimum eccentricity.

Assume now that tree T is rooted. The *depth* of T ($\mathsf{depth}\ T$) is the eccentricity of the root of T. A *depth path* of T is a path from the root to a leaf with maximum distance from the root. The *depth tree* of T is the union of all the depth paths of T. A *diametral path* of T is a path between two leaves with maximum eccentricity. The *diametral tree* of a tree is the union of all the diametral paths of a tree.

Let Γ be an embedding of a planar connected graph G, and let T^* be the block-cutface tree of Γ. We define the *diameter* of Γ ($\mathsf{diam}\ \Gamma$) as the diameter of T^*. If Γ is a planar embedding, we define the *depth* of Γ ($\mathsf{depth}\ \Gamma$) as the depth of T^*.

We consider the problem of assembling the planar embedding of a connected planar graph G using given embeddings for its blocks, with the goal of minimizing the depth of the embedding.

Let G be a connected planar graph, and assume that we have a prescribed embedding for each block B of G. An embedding Γ of G is said to *preserve the embedding of a block* B if the sub-embedding of B in Γ is the same as the prescribed embedding of B. We say that Γ is *block-preserving for a cutvertex* v if Γ preserves the embedding of each block B containing v, that is, the circular order of the edges of B incident on v is equal to the order in the prescribed embedding of B. Finally, we say that Γ is *block-preserving* if it preserves the embedding of each block. This is equivalent to saying that it is block-preserving for each cutvertex.

We are now ready to define formally the problem studied in this paper:

Problem 1 (Depth Minimization). Given a planar connected graph G and an embedding for each block of G, compute a block-preserving planar embedding Γ of G with minimum depth. □

Solving Problem 1 requires choosing the external face of the planar embedding. We shall find it convenient to study first a restricted version of the problem where the external face must contain a given cutvertex.

Problem 2 (Constrained Depth Minimization). Given a planar connected graph G, an embedding for each block of G, and a cutvertex v of G, compute a block-preserving planar embedding Γ of G, such that v is on the external face and Γ has minimum depth. □

Note that algorithms for Problems 1 and 2 need to access only the circular ordering of the edges around the cutvertices in the prescribed embedding of each block. We now introduce further definitions in order to simplify the description of the solution space of the problems.

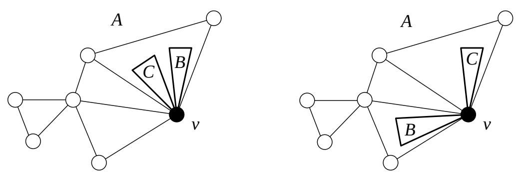
Given a cutvertex v of G and block B containing v, we call the pair (B, v) a *cutpair*. The faces of block B containing v are called the *candidate cutfaces* for the cutpair (B, v) since one or more of them will be a cutface in block preserving embedding of the entire graph G.

Lemma 1 *Given a connected planar graph G with a prescribed embedding for its blocks, there exists a block-preserving planar embedding Γ of G* with minimum depth *such that, for each cutpair (B, v) of G, all the blocks incident to v distinct from B are nested inside the same candidate cutface of (B, v).*

Proof. Let Γ be a minimum-depth block-preserving embedding of graph G. For every cutvertex v of G, if there is more than one cutface incident on v, we choose f among the cutfaces incident on v with minimum distance from the external face. We move all the blocks that are not incident to f in it. This operation can only shorten the distance from the moved blocks to the external face in the block-cutface tree of Γ. Thus, the minimality of the embedding is preserved. □

Motivated by Lemma 1, and observing that the order of the blocks around a cutvertex does not affect the depth of a planar embedding, we represent a block-preserving embedding with the selection of a candidate cutface for each cutpair (see Figure 3):

Definition 2 *Let G be a connected planar graph G with a prescribed embedding for its blocks. A* nesting *for G is the assignment of candidate cutface f to each each cutpair (B, v). A nesting of G describes a class of planar embeddings of G with the same depth.*



(a) (b)

Fig. 3. Two embeddings of the blocks around a cutvertex v: (a) embedding described by a nesting with a single cutface; (b) embedding with two cutfaces that is not described by a nesting. Note that the embedding of part (a) has the same depth as the one of part (b).

Using the concept of nesting, Lemma 1 can be restated as follows:

Lemma 3 *Given a connected planar graph G with a prescribed embedding for its blocks, there exists a planar embedding of G with minimum depth that is described by a nesting.*

3 Constrained Depth Minimization

In this section, we present Algorithm 4 (*ConstrainedMinimization*) for solving Problem 2 (Constrained Depth Minimization). It takes as input a connected planar graph G with a prescribed embedding for its blocks, and a cutvertex v. The output is an embedding Γ of G that has v on the external face and minimizes $\mathsf{depth}\ \Gamma$. The algorithm considers the block-cutvertex tree T of G rooted at v and builds the planar embedding Γ by means of a post-order traversal of T.

Given a node x of T (it may be a cutvertex or a block), we denote with $G(x)$ the subgraph of G associated with the subtree of T rooted at x. We denote with $\Gamma(x)$ the planar embedding of $G(x)$ computed by method $\mathsf{embed}(x)$ of the algorithm. Method $\mathsf{embed}(x)$ takes as input graph $G(x)$ and returns a planar embedding $\Gamma(x)$ of $G(x)$ with minimum depth. Let $y_1, y_2, \cdots, y_m$ be the children of x in T. The embedding $\Gamma(x)$ restricted to graph $G(y_i)$ is the embedding $\Gamma(y_i)$ returned by $\mathsf{embed}(y_i)$. In other words, we build the embedding $\Gamma(x)$ by assembling the previously computed embeddings $\Gamma(y_i)$ of the children of x.

We describe the embeddings $\Gamma(x)$ and $\Gamma(y_1), \cdots, \Gamma(y_m)$ by means of nestings (see Definition 2). When building $\Gamma(x)$, the information provided by $\Gamma(y_i)$'s is not enough. If x is a cutvertex, we need to select the candidate cutface for cutpairs (y_i, x), $i = 1, \cdots, m$, and if x is a block, we need to select the candidate cutface for cutpairs (x, y_i), $i = 1, \cdots, m$.

We prove the correctness of the algorithm with an inductive argument. The base case is for leaf blocks of T: Given a leaf block B of T, the algorithm correctly returns as $\Gamma(B)$ the prescribed embedding of B, which is trivially minimal.

If B is a non-leaf block of T, we assume, by the inductive hypothesis, the minimality of the planar embeddings of the children of B. We now consider the embedding $\Gamma(B)$ computed by method $\mathsf{embed}(v)$ and the block-cutface tree

Algorithm 4 ***ConstrainedMinimization***

input A connected planar graph G with block-cutvertex tree T, a prescribed embedding for the blocks of G, and a cutvertex v of G.
output A block-preserving planar embedding Γ of G of minimum depth with v on the external face.

The algorithm computes and returns $\Gamma = \mathsf{embed}(v)$.

method $\mathsf{embed}(v)$

input A cutvertex v of T.
output A block-preserving planar embedding $\Gamma(v)$ of minimum depth with v on the external face.

```
for all children B of v in T do
  if B is a leaf block (i.e., it has no children) then
    Let Γ(B) be the prescribed embedding of block B, with external face
    equal to one of the candidate cutfaces of (B, v).
  else
    Let Γ(B) = embed(B).
  end if
end for
for all children B of v in T do
  Assign the external face of Γ(B) to cutpair (B, v).
end for
Construct planar embedding Γ(v) by joining the previously computed pla-
nar embeddings of the children blocks of v and placing them on the external
face.
```

method $\mathsf{embed}(B)$
input A block B of T.
output A block-preserving planar embedding $\Gamma(B)$ of minimum depth with external face equal to one of the candidate cutfaces of cutpair $(B, \text{parent}(B))$.

for all children v of B in T **do**
  Let $\Gamma(v) = \mathsf{embed}(v)$.
end for
for all candidate cutfaces f of $(B, \text{parent}(B))$ **do**
  Let $\delta(f)$ be the maximum depth of the (previously computed) planar embeddings $\Gamma(v)$ of the cutvertices children of B on face f.
end for
Let δ_B be the maximum of $\delta(f)$ over all candidate cutfaces of $(B, \text{parent}(B))$. Let f_B be a candidate cutface of $(B, \text{parent}(B))$ such that $\delta(f_B) = \delta_B$. If more than one such cutfaces exist, choose one with the maximum number of cutvertices with deepest embedding (i.e., cutvertices w such that $\Gamma(w)$ has depth δ_B).
for all children v of B in T **do**
  if v is on face f_B **then**
    Assign cutface f_B to cutpair (B, v).
  else
    Assign an arbitrary cutface to cutpair (B, v).
  end if
end for
Construct planar embedding $\Gamma(B)$ by adding to the prescribed embedding of B the previously computed planar embeddings $\Gamma(v)$ of the children of B, where, for a child v of B, embedding $\Gamma(v)$ is placed inside the cutface assigned to (B, v).
Let f_B be the external face of $\Gamma(B)$.

$T^*(B)$, which is rooted at the external face f_B (which is a cutface of the entire embedding Γ. The children of f_B in $T^*(B)$ are blocks, one of which is B itself. We can write $\mathsf{depth}\ \Gamma(B)$ in the following way:

$$\mathsf{depth}\ \Gamma(B) = \max\{d_P, d_{NP}\}$$

where

$$d_P = \max_{v \text{ incident to } f_B} \mathsf{depth}\ \Gamma(v)$$

and

$$d_{NP} = 2 + \max_{v \textbf{ not} \text{ incident to } f_B} \mathsf{depth}\ \Gamma(v)$$

where "incident" refers to the embedding $\Gamma(B)$.

The quantity d_P represents the contribution of the planar embeddings whose external face is f_B in $T^*(B)$, we say that such embeddings are *promoted*. The quantity d_{NP} represents the contribution of the planar embeddings whose external face is a child of B in $T^*(B)$ (see Figure 4).

The minimality of d_P follows from the fact that each $\Gamma(v)$ is minimal. The minimality of d_{NP} can be shown as follows: for any child v of B, the length of a root-to-leaf path in $T^*(B)$ is at most $\Gamma(v) + 2$. The algorithms selects the external face f_B of $\Gamma(B)$ that optimizes the depth by trying to place the deepest sub-embeddings of the children of B on the external face.

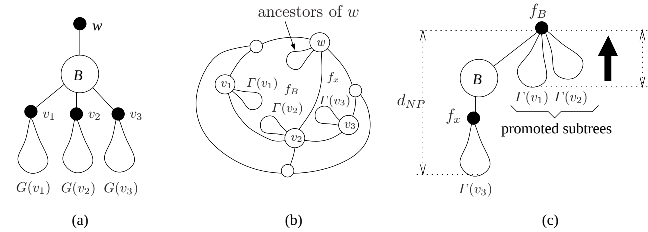


Fig. 4. (a) A block-cutvertex tree. (b) Embedding $\Gamma(B)$ computed by method $\mathsf{embed}(B)$. (c) Block cutface-tree of $\Gamma(B)$, where $\Gamma(v_1)$ and $\Gamma(v_2)$ are promoted, while $\Gamma(v_3)$ is not.

Given a cutvertex v, we assume, by the inductive hypothesis, the minimality of the planar embeddings $\Gamma(B)$ of the children of v computed by $\mathsf{embed}(B)$. It is easy to see that

$$\mathsf{depth}\ \Gamma(v) = \max_{B \in \text{children}(v)} \mathsf{depth}\ \Gamma(B)$$

where children(v) denotes the set of children of v in T. The minimality of $\Gamma(v)$ follows directly from the minimality of $\mathsf{depth}\ \Gamma(B)$ for each child B of v.

The running time of all the executions of method $\mathsf{embed}(B)$, excluding the calls to method $\mathsf{embed}(v)$, can be written as $\sum_{B \in T} \ \sum_{f \in B} \ |f|$ where $|f|$ denotes the number of edges of face f.

The running time of all the executions of method $\mathsf{embed}(v)$, excluding the calls to method $\mathsf{embed}(B)$, can be written as $\sum_{v \in T} \ \sum_{B \in \text{children}(v)} \ |f_B|$ where $|f_B|$ denotes the number of edges of the external face f_B of B.

Each of the above sums can be shown to be equal to the number of edges of a planar graph with $O(n)$ edges, where n is the number of vertices of G. Thus, we conclude that the overall running time of Algorithm 4 (*ConstrainedMinimization*) is $O(n)$.

The main result of this section is summarized in the following theorem:

Theorem 5 *Given a planar connected graph G with n vertices, a prescribed embedding for each block of G, and a cutvertex v of G, Algorithm 4 (*ConstrainedMinimization*) computes in $O(n)$ time a block-preserving planar embedding Γ of G, such that v is on the external face and Γ has minimum depth.*

4 General Depth Minimization

A brute-force quadratic-time algorithm for Problem 2 (Depth Minimization) consists of repeatedly executing Algorithm 4 (ConstrainedMinimization) for each cutvertex, and then returning the planar embedding with minimum depth.

In this section, we present a linear-time algorithm for depth minimization based on the following approach:

1. compute an embedding with the minimum diameter;
2. select the external face to minimize the depth of the resulting planar embedding.

The reduction used in our approach is summarized in the following theorem:

Theorem 6 *Given a connected planar graph G whose blocks have a prescribed embedding, and a block-preserving embedding Δ of G with minimum diameter, a block-preserving planar embedding of G with minimum depth is obtained from Δ by selecting the external face of Δ as a face with minimum eccentricity in the block-cutface tree of Δ.*

By Theorem 6, we can solve Problem 1 by making an appropriate choice of the external face, provided an efficient algorithm for minimizing the diameter of an embedding is given. Before proving Theorem 6, we state two basic properties of the eccentricity of the nodes of a block-cutvertex tree.

Property 7 *In a block-cutvertex tree, each block-node has even eccentricity while each cutvertex-node has odd eccentricity.*

Property 8 *The center of a block-cutvertex tree contains only one node.*

Since any block-cutface tree is the block-cutvertex tree of the dual embedding of a graph, the above properties hold also for block-cutface trees, where any occurrences of the word cutvertex is replaced by cutface.

In the following, we denote the eccentricity of a face f of an embedding Γ in the block-cutface tree of Γ with $e_\Gamma(f)$. Similarly, we denote with $e_\Gamma(B)$ the eccentricity of a block B.

Lemma 9 *Let Γ be an embedding of a connected planar graph, and let f be a face of Γ with minimum eccentricity. We have:*

$$2 \cdot e_\Gamma(f) - 2 \leq \mathsf{diam}\ \Gamma \leq 2 \cdot e_\Gamma(f)$$

Proof. If f is the center of the block-cutface tree then it is in the middle of a diametral path of even length, thus $\mathsf{diam}\ \Gamma = 2 \cdot e_\Gamma(f)$.

If f is adjacent to the center B (a block) of the block-cutface tree, we have $\mathsf{diam}\ \Gamma = 2 \cdot e_\Gamma(B)$, and hence $\mathsf{diam}\ \Gamma = 2 \cdot (e_\Gamma(f) - 1)$. □

We are now ready to prove Theorem 6.

Proof. (of Theorem 6)

We show that, if there exists a planar embedding Γ with external face f_Γ that has smaller depth than the embedding Δ with external face f_Δ, then Δ does not have minimum diameter, which is a contradiction.

Suppose that $\mathsf{depth}\ \Gamma < \mathsf{depth}\ \Delta$. Since the eccentricities of cutfaces are always odd, the difference between the two depths is at least 2, i.e., $e(f_\Gamma) \leq e(f_\Delta) - 2$. Hence, by Lemma 9, we obtain

$$\mathsf{diam}\ \Gamma \leq 2 \cdot e(f_\Gamma) \leq 2 \cdot e(f_\Delta) - 4 < 2 \cdot e(f_\Delta) - 2 \leq \mathsf{diam}\ \Delta.$$

□

We say that a planar embedding Γ has *total minimum depth* if $\Gamma(x)$ has minimum depth for each node x of its block-cutface tree. The embedding produced by Algorithm 4 (ConstrainedMinimization) has total minimum depth. We show that this embedding either has optimal diameter, or can be easily transformed into an embedding with optimal diameter.

Intuitively, much of the minimization work performed by Algorithm 4 may be reused to minimize the diameter. Since the algorithm recursively minimizes the depth, each diametral path spans one or two subtrees with minimum depth. If any depth reduction is possible, this should be at the root of such subtrees.

Consider a tree T with root r and its diametral tree T_{diam}. We call *knot* of T the node of T_{diam} closest to r. As a special case, the knot may coincide with r. It is easy to prove that all the diametral paths of T contain the knot.

Theorem 10 *Let G be a connected planar graph with prescribed embeddings for its blocks. Given a planar embedding Γ of G with total minimum depth, there exists a minimum diameter embedding Δ of G such that either*

- *$\Delta = \Gamma$; or*
- *Δ differs from Γ at most for the embeddings of the cutvertices around one block, which is the knot of the block-cutface tree of Γ, and we have* $\mathsf{diam}\ \Delta = \mathsf{diam}\ \Gamma - 2$.

Proof. (sketch) Consider the block-cutface tree T^* of Γ, which is rooted at the external face, and the block-cutvertex tree T of G. We call T_{diam} the diametral tree of T^*, T_{depth} its depth tree, and k its knot. We consider T_{diam} rooted at k, where k may be a block or a cutface. We distinguish two cases:

- *k is a cutface.* In this case, we set $\Delta = \Gamma$, which has minimum diameter. Indeed, any diametral path of T^* may be written as $B_a, \dots, B_b, k, B_c, \dots, B_d$, where B_b and B_c are children of k in $T_{diam}(k)$. The subpaths $B_a, \dots, B_b$ and $B_c, \dots, B_d$ have minimum length because Γ has total minimum depth and thus $\Gamma(B_b)$ and $\Gamma(B_c)$ are minimum depth embedding of $G(B_b)$ and $G(B_c)$ respectively. Note that selecting different cutfaces for the cutpairs (B_b, k) and (B_c, k) does not change the distance between B_b and B_c.
- *k is a block.* If k is the root of T_{depth}, then $T_{diam} = T_{depth}$ and we set $\Delta = \Gamma$, which has minimum diameter. Else, any diametral path of T^* may be written as $B_a, \dots,\ f_a, k,\ f_b, \dots,\ B_b$. The subpaths $B_a, \dots, f_a$ and $f_b, \dots, B_b$ have minimum length because Γ has total minimum depth. If we can select the same candidate cutface of the cutpair (k, w) for all the cutvertices w that are children of k in T_{diam}, we obtain an embedding Δ such that $\mathsf{diam}\ \Delta = \mathsf{diam}\ \Gamma - 2$. Else, we set $\Delta = \Gamma$. □

Our algorithm for finding a minimum diameter embedding consists of the following steps.

1. Use Algorithm 4 to construct an embedding Γ with total minimum depth with respect to an arbitrary cutvertex.

2. Find the knot of the block-cutface tree of Γ and, if needed, modify Γ into a minimum diameter embedding as described in the proof of Theorem 10.
3. Apply Theorem 6 to select an external face that minimizes the depth of the resulting planar embedding.

Note that a planar embedding obtained with the above algorithm has minimum depth, but in general does not have total minimum depth. To obtain the latter, we run Algorithm 4 again with a starting cutvertex on the external face.

We summarize the main result of this paper in the following theorem:

Theorem 11 *Given a planar connected graph G with n vertices and a prescribed embedding for each block of G, a block-preserving planar embedding of G with minimum depth can be computed in $O(n)$ time.*

5 Conclusions

We have presented an optimal algorithm for the problem of minimizing the depth of a planar embedding whose blocks have a prescribed embedding. Our results have practical applications to graph drawing systems. Future work includes:

– Investigate a weighted version of the minimum-depth embedding problem, where the blocks have positive weights and we want to minimize the weighted depth of the block-cutface tree. This variation of the problem is relevant in graph drawing applications, where the size of the drawings of the blocks should be taken into account.
– Perform an extensive experimental study on the effect of using our depth minimization technique in graph drawing algorithms based on the topology-shape-metric approach.

References

1. D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD-Library: A library of algorithms for graph drawing. In *Proc. Workshop on Algorithm Engineering*, pages 112–123, 1997. 357
2. C. Batini, E. Nardelli, and R. Tamassia. A layout algorithm for data-flow diagrams. *IEEE Trans. Softw. Eng.*, SE-12(4):538–546, 1986. 357
3. C. Batini, M. Talamo, and R. Tamassia. Computer aided layout of entity-relationship diagrams. *Journal of Systems and Software*, 4:163–173, 1984. 357
4. G. D. Battista, W. Didimo, M. Patrignani, and M. Pizzonia. Orthogonal and quasi-upward drawings with vertices of prescribed size. In *Graph Drawing (Proc. GD '99)*, volume 1731 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1999. 357
5. P. Bertolazzi, G. Di Battista, and W. Didimo. Computing orthogonal drawings with the minimum number of bends. In *Workshop Algorithms Data Struct. (WADS'97)*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 331–344. Springer-Verlag, 1997. 358
6. D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM Journal on Computing*, 17(1):53–76, Feb. 1988. 357

7. D. Bienstock and C. L. Monma. Optimal enclosing regions in planar graphs. *Networks*, 19:79–94, 1989. 357
8. D. Bienstock and C. L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990. 357
9. U. Brandes and D. Wagner. A Bayesian paradigm for dynamic graph layout. In *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 236–247. Springer-Verlag, 1998. 357
10. U. Brandes and D. Wagner. Dynamic grid embedding with few bends and changes. *Lecture Notes in Computer Science*, 1533, 1998. 357
11. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999. 357
12. G. Di Battista et al. *Graph Drawing Toolkit.* University of Rome III, Italy. http://-www.dia.uniroma3.it/~gdt/. 357
13. G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7:303–325, 1997. 357
14. G. Di Battista, G. Liotta, and F. Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998. 358
15. W. Didimo and G. Liotta. Computing orthogonal drawings in a variable embedding setting. In *Algorithms and Computation (Proc. ISAAC '98)*, volume 1533 of *Lecture Notes Comput. Sci.*, pages 79–88. Springer-Verlag, 1998. 358
16. U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes Comput. Sci.*, pages 254–266. Springer-Verlag, 1996. 357
17. A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Computing.* to appear. 358
18. A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 286–297. Springer-Verlag, 1995. 358
19. J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974. 357
20. M. Jünger, S. Leipert, and P. Mutzel. Pitfalls of using PQ-Trees in automatic graph drawing. In *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 193–204. Springer-Verlag, 1997. 357
21. M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16(1):33–59, 1996. 357
22. G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In *IPCO: 7th Integer Programming and Combinatorial Optimization Conference*, volume 1610 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1999. 357
23. H. Lauer, M. Ettrich, and K. Soukup. GraVis - system demonstration. In *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 344–349. Springer-Verlag, 1997. 357
24. P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph. In *IPCO: 7th Integer Programming and Combinatorial Optimization Conference*, volume 1610 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1999. 357
25. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987. 357
26. R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988. 357