Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Computing a Minimum-Depth Planar Graph Embedding in $O(n^4)$ Time

Patrizio Angelini, Giuseppe Di Battista, and Maurizio Patrignani

RT-DIA-116-2007 **May 2007**

Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
{angelini,gdb,patrigna}@dia.uniroma3.it

Work partially supported by EC - Fet Project DELIS - Contract no 001907 and by MUR under Project "MAINSTREAM: Algoritmi per strutture informative di grandi dimensioni e data streams".

ABSTRACT

Consider an n-vertex planar graph G. We present an $O(n^4)$-time algorithm for computing an embedding of G with minimum distance from the external face. This bound improves on the best previous bound by an $O(n \log n)$ factor. As a side effect, our algorithm improves the bounds of several algorithms that require the computation of a minimum distance embedding.

1 Introduction

As pointed out in [3, 8, 7], the quality of a planar embedding of a planar graph can be measured in terms of maximum distance of its vertices from the external face. Such a distance can be given in terms of different incidence relationships between vertices and faces. For example, if we say that two faces are adjacent if they share a vertex, the maximum distance to the external face is called *radius* [9]. If two vertices are adjacent if they are endpoints of an edge, the maximum distance to the external face is called *width* [6]. If two vertices are adjacent if they are on the same face and the external face is adjacent to all its vertices, the maximum distance to the external face is called *outerplanarity* [1]. If two faces are adjacent if they share an edge, the maximum distance to the external face is called *depth* [2]. Fig. 1 shows two embeddings of the same graph where internal faces are labeled with their distance from the external face.

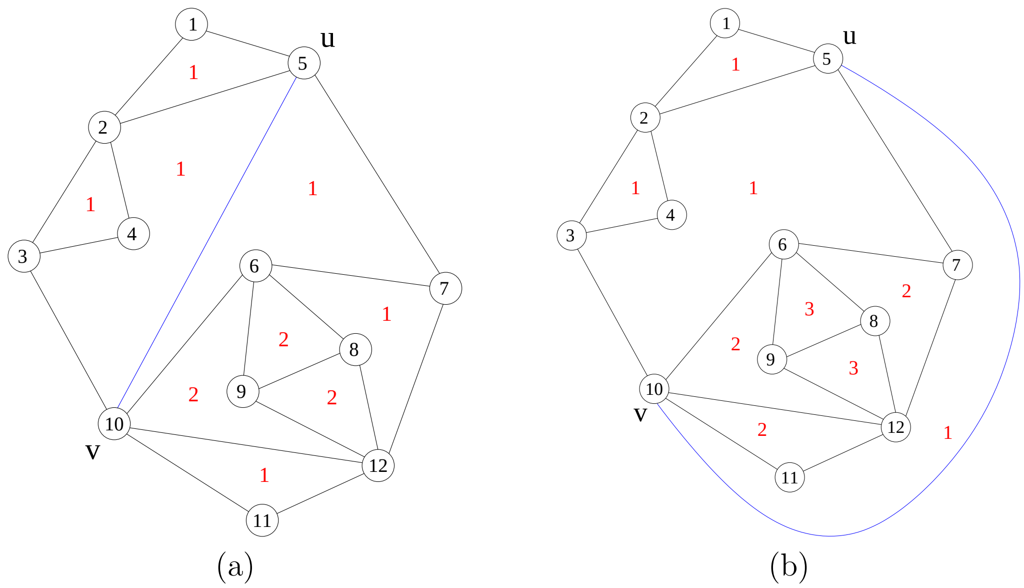


Figure 1: Two embeddings of the same graph G lead to different values of depth. Internal faces are labeled with their distance from the external face.

The algorithms that compute a planar embedding such that the vertices have minimum maximum distance to the external face have several applications. Let us give a few examples. The algorithm by Dolev, Leighton, and Trickey for drawing planar graphs with asymptotically optimal area [6] requires the computation of the embedding with minimum width. In [1] Baker gives approximation algorithms on planar graphs for many NP-complete problems, including maximum independent set and minimum vertex cover. The time complexity and the optimality bounds of such algorithms depend on the outerplanarity of the graph. Finally, the algorithm by Di Giacomo et al. [5] for constructing crossing-free minimum radial drawings of planar graphs is based on the computation of their outerplanarity.

In [3] Bienstock and Monma present an algorithm to compute the planar embedding of an n-vertex planar graph G with minimum maximum distance to the external face in $O(n^5 \log n)$ time. The distance they consider is the depth. However, it is not difficult to compute the other distances listed above with variations of the algorithm. The algorithm is based on the decomposition of the graph into its biconnected and triconnected components. The general approach is the one of selecting a positive integer k and to check if an embedding exists with depth k. A binary search is done to determine the optimal value of k. For each selected k the decomposition of G is visited associating to each component μ a left and a right weight, corresponding to the distances of the left and right border of μ from the external face of G. Such weights are independent on the embedding of the components. The components are then visited to check if their weights can be composed to construct an embedding with depth k. The space complexity of the algorithm is not analyzed in the paper.

In [8] Pizzonia and Tamassia present an algorithm for solving in $O(n)$ time an analogous problem where the depth of the embedding is expressed in terms of biconnected components traversed to reach the external face and the biconnected components are "rigid", in the sense that their embedding cannot be changed.

In this paper we present an algorithm that improves the time bound of [3] to $O(n^4)$ time. As a side effect, we improve also the time bound of the algorithms listed above that need to compute a planar embedding with minimum maximum distance to the external face. We concentrate the attention on the depth; however, as was pointed out in [3], computing the other distances can be done with variations of the algorithm.

Our approach is inspired by the methods in [3], and develops on top of such methods several new techniques. As in [3] we decompose the graph into bi- and tri-connected components, using BC-trees and SPQR-trees [4]. However, we are able to solve the problem on each biconnected component, with a given edge on the external face, in $O(n^3)$ time. Then, we use techniques analogous to those in [3] for assembling the results on each biconnected component into a general solution. Among the techniques presented in this paper, a key issue, that might have other applications, is the ability of representing implicitly and with reasonable size all the possible values of depth of each triconnected component. The space complexity of the algorithm is $O(n^3)$.

The paper is organized as follows. Section 2 gives basic definitions. Section 3 deals with the combinatorial structure of the depth of the planar embeddings and develops a theory of the set of integer pairs that is exploited in the algorithm. Section 4 presents the algorithm for biconnected graphs and Section 5 extends the algorithm to general connected graphs. In Section 6 we give concluding remarks and further compare our approach with the one in [3].

2 Background

A graph $G(V, E)$ is *connected* if every pair of vertices of G is connected by a path. A *separating k-set* of a graph G is a set of k vertices whose removal increases the number of connected components of G. Separating 1-sets and 2-sets are called *cutvertices* and *separation pairs*, respectively. A connected graph is *biconnected* if it has no cutvertices. The maximal biconnected subgraphs of a graph are its *blocks*. Observe that each edge of G falls into a single block of G, while cutvertices are shared by different blocks. The *block cutvertex tree*, or BC-tree, of a connected graph G has a B-node for each block of G and a C-node for each cutvertex of G. Edges in the BC-tree connect each B-node μ to the C-nodes associated with the cutvertices in the block of μ. The BC-tree of G may be thought as rooted at a specific block ν.

A *split pair* $\{u, v\}$ of a graph G is either a separation pair or a pair of adjacent vertices. A *maximal split component* of G with respect to a split pair $\{u, v\}$ (or, simpler, a maximal split component of $\{u, v\}$) is either an edge (u, v) or a maximal subgraph G' of G such that G' contains u and v and $\{u, v\}$ is not a split pair of G'. A vertex w distinct from u and v belongs to exactly one maximal split component of $\{u, v\}$. We call *split component* of $\{u, v\}$ a subgraph of G that is the union of any number of maximal split components of $\{u, v\}$.

In the following, we summarize SPQR-trees. For more details, see [4]. An example of an SPQR-tree is shown in Fig. 2. SPQR-trees are closely related to the classical decomposition of biconnected graphs into triconnected components. Let $\{s, t\}$ be a split pair of G. A *maximal split pair* $\{u, v\}$ of G with respect to $\{s, t\}$ is a split pair of G distinct from $\{s, t\}$ such that, for any other split pair $\{u', v'\}$ of G, there exists a split component of $\{u', v'\}$ containing vertices u, v, s, and t. Let $e = (s, t)$ be an edge of G, called *reference edge*. The SPQR-tree $\mathcal{T}$ of G with respect to e describes a recursive decomposition of G induced by its split pairs. Tree $\mathcal{T}$ is a rooted ordered tree whose nodes are of four types: S, P, Q, and R. Denote by G' the st-biconnectible graph obtained from G by removing e.

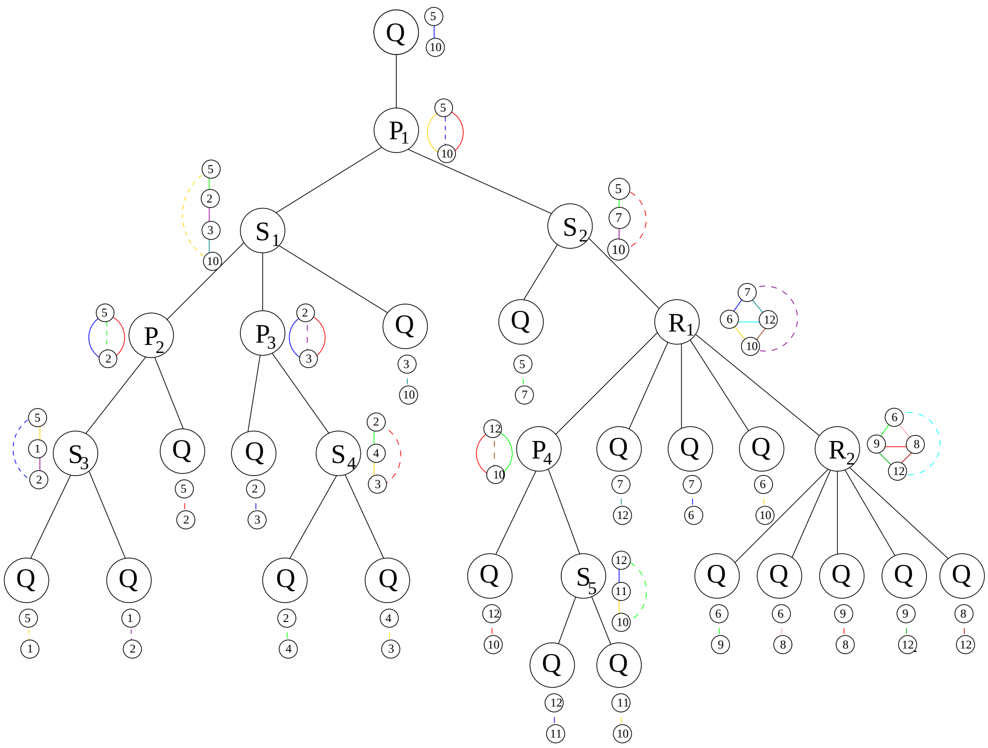


Figure 2: SPQR tree of the graph of Fig. 1 with $(5, 10)$ as reference edge.

Each node μ of $\mathcal{T}$ has an associated st-biconnectible multigraph, called the *skeleton* of μ and denoted by $sk(\mu)$. Also, each node μ is associated with an edge of the skeleton of the parent ν of μ, called the *virtual edge* of μ in $sk(\nu)$. The skeleton $sk(\mu)$ shows how its $\delta(\mu)$ lesser components, represented in their turn by *virtual edges*, are arranged into the current component.

Starting from a virtual edge (u, v) of $sk(\mu)$, or from the skeleton of the corresponding child node ν, by recursively replacing each virtual edge with the skeleton of the corresponding component, it can be obtained a subgraph of G called the *pertinent graph* of (u, v) and denoted by $pertinent(\nu)$.

Tree $\mathcal{T}$ is recursively defined as follows.

Trivial Case: If G consists of exactly one edge between s and t, then $\mathcal{T}$ consists of a single Q-node whose skeleton is G itself.

Parallel Case: If the split pair $\{s, t\}$ has at least two maximal split components $G_1, \ldots, G_k$ $(k \geq 2)$, the root of $\mathcal{T}$ is a P-node μ. Graph $sk(\mu)$ consists of k parallel edges between s and t, denoted by $e_1, \ldots, e_k$.

Series Case: If the split pair $\{s, t\}$ has exactly one maximal split component G' which is not a single edge and if G' has cutvertices $c_1, \ldots, c_{k-1}$ $(k \geq 2)$ in this order on a path from s to t, the root of $\mathcal{T}$ is an S-node μ. Graph $sk(\mu)$ is the path $e_1, \ldots, e_k$, where e_i connects c_{i-1} with c_i $(i = 2 \ldots k-1)$, e_1 connects s with c_1, and e_k connects c_{k-1} with t.

Rigid Case: If none of the above cases applies, let $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$ be the maximal split pairs of G with respect to $\{s, t\}$ $(k \geq 1)$ and, for $i = 1, \ldots, k$, let G_i be the union of all the maximal split components of $\{s_i, t_i\}$. The root of $\mathcal{T}$ is an R-node μ. Graph $sk(\mu)$ is the triconnected graph obtained from G by replacing each subgraph G_i with the edge e_i between s_i and t_i.

Except for the trivial case, μ has children $\mu_1, \ldots, \mu_k$, in this order, such that μ_i is the root of the SPQR-tree of graph $G_i \cup (u_i, v_i)$ with respect to reference edge (u_i, v_i) $(i = 1, \ldots, k)$. Edge (u_i, v_i) is said to be the *virtual edge* of node μ_i in $sk(\mu)$ and of node μ in $sk(\mu_i)$. Graph G_i is called the *pertinent graph* of node μ_i and of edge (u_i, v_i) and it is denoted by pertinent(u_i, v_i). Vertices u and v are the *poles* of G_i. The poles are always placed on the external face of the skeleton.

The tree $\mathcal{T}$ so obtained has a Q-node associated with each edge of G, except the reference edge e. We complete the SPQR-tree $\mathcal{T}$ by adding another Q-node, representing the reference edge e, and making it the parent of μ so that it becomes the root of $\mathcal{T}$.

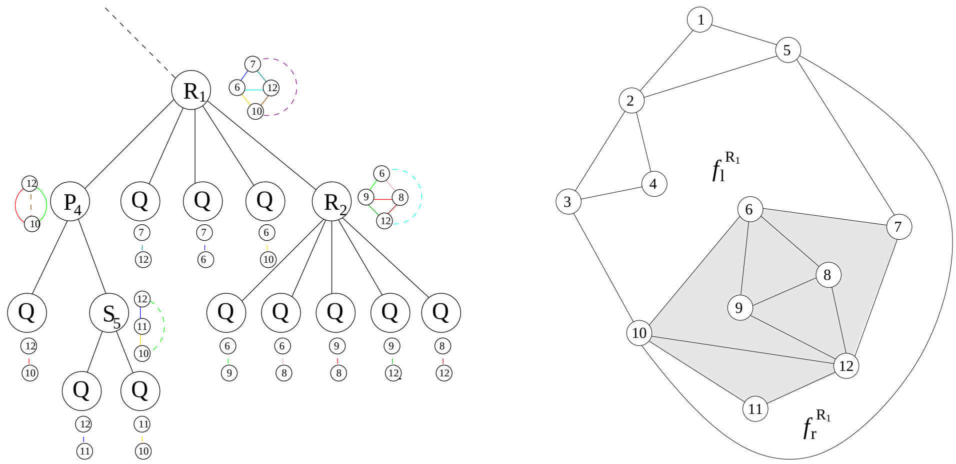
The SPQR-tree $\mathcal{T}$ of a graph G with n vertices and m edges has m Q-nodes and $O(n)$ S-, P-, and R-nodes. Also, the total number of vertices of the skeletons stored at the nodes of $\mathcal{T}$ is $O(n)$.

3 The Combinatorial Structure of Planar Embeddings and their Depths

A biconnected graph G is planar if and only if the skeletons of all the nodes of its SPQR-tree are planar. An SPQR-tree $\mathcal{T}$ rooted at a given Q-node can be used to represent all the planar embeddings of G having the reference edge associated with the Q-node at the root on the external face. Namely, any embedding can be obtained by selecting one of the two possible flips of each skeleton around its poles and selecting a permutation of the children of each P-node with respect to their common poles.

Let G be a biconnected planar graph and let μ_i be a component of the SPQR-tree decomposition $\mathcal{T}$ of G rooted at edge e. Observe that any embedding Γ_G of G with e

on the external face corresponds to an embedding $\Gamma_{G_{\mu_i}}$ of the pertinent graph G_{μ_i} of μ_i with poles u_i and v_i on the external face. Also, the external face of $\Gamma_{G_{\mu_i}}$ corresponds to two faces of Γ_G, which can be arbitrarily called *left and right external faces of* G_{μ_i} and denoted by $f_l^{\mu_i}$ and $f_r^{\mu_i}$. For example, Fig. 3.a shows the subtree rooted at an R-node R_1 of the SPQR-tree decomposition of a graph G, and Fig. 3.b shows an embedding of G. The embedding of G induces an embedding of the pertinent graph of R_1 (shown in grey), whose external face corresponds to the two faces labeled $f_l^{R_1}$ and $f_r^{R_1}$.



(a) (b)

Figure 3: (a) The subtree rooted at node R_1 of the SPQR-tree represented in Fig. 2. (b) The two faces $f_l^{R_1}$ and $f_r^{R_1}$ of the embedding of G that correspond to the external face of R_1.

Following the approach of [3], we use a definition of faces $f_l^{\mu_i}$ and $f_r^{\mu_i}$ which is independent on the embedding Γ_G and only depends on $\Gamma_{G_{\mu_i}}$. Let (u_i, v_i) be the virtual edge of μ_i that represents in μ_i the portion of G containing e and denote by $G^+_{\mu_i}$ the graph obtained by adding edge (u_i, v_i) to the pertinent graph G_{μ_i} of μ_i. Suppose $G^+_{\mu_i}$ is planarly embedded. The (u_i, v_i)-*dual* of G_{μ_i} is obtained by computing the dual of $G^+_{\mu_i}$ and removing the edge of the dual corresponding to (u_i, v_i). The faces incident to the removed edge are $f_l^{\mu_i}$ and $f_r^{\mu_i}$. Fig. 4 shows the construction of the (u, v)-dual graph of component R_2.

A component μ_i *satisfies* the pair of non-negative integers $\langle x, y\rangle$ if its pertinent graph G_{μ_i} admits an embedding $\Gamma_{G_{\mu_i}}$, with its poles on the external face, where it is possible to find a partition of the set of its internal faces into two sets, denoted by F_l and F_r, such that all faces in F_l have distance from $f_l^{\mu_i}$ less or equal than x and all faces in F_r have distance from $f_r^{\mu_i}$ less or equal than y. Fig. 5 shows an embedding of component S_2 of the SPQR-tree of Figure 2 and three possible partitions of its faces corresponding to integer pairs $\langle 0, 2\rangle$, $\langle 1, 2\rangle$, and $\langle 2, 0\rangle$. Obviously, if μ_i satisfies $\langle x, y\rangle$, then it satisfies any pair $\langle w, z\rangle$ with $w \geq x$ and $z \geq y$. The infinite set of integer pairs satisfied by component μ_i is the *admissible set of* μ_i, and is denoted by $A(\mu_i)$.

In order to efficiently represent the admissible set of μ_i, we need to investigate its combinatorial properties. Hence, we provide a definition of a "precedence" relationship between integer pairs and explore the combinatorial properties of sets of integer pairs. Such properties can be also expressed in terms of poset theory or inclusion relationships

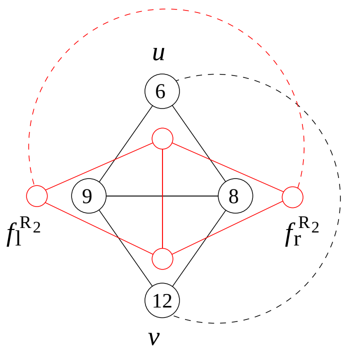


Figure 4: (u, v)-dual graph of component R_2. Pertinent graph of R_2 is drawn black and its (u, v)-dual graph is drawn red. Virtual edge (u, v) and its dual are drawn dashed.

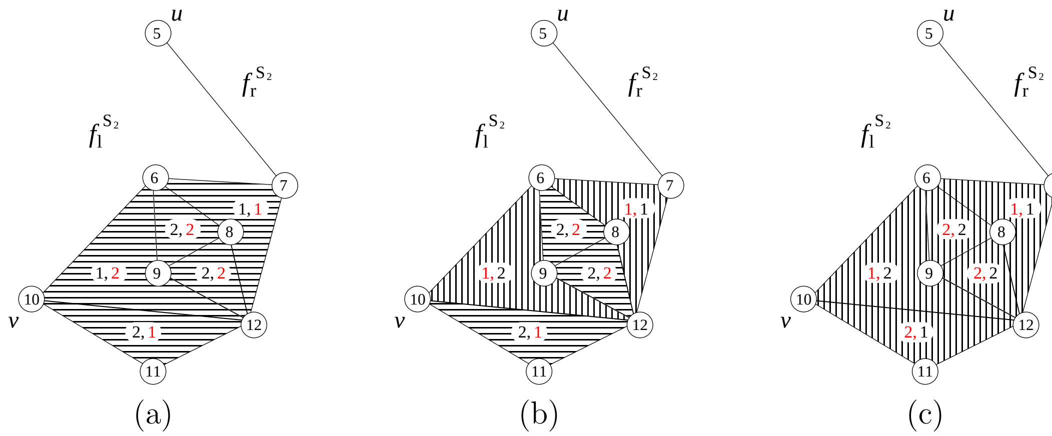


Figure 5: An embedding of component S_2 of the SPQR-tree of Figure 2. Figures (a), (b), and (c) represent three possible partitions of the internal faces. Faces filled with vertical lines belong to F_l and faces filled with horizontal lines belong to F_r. Each face is labeled with two integers representing its distance from $f_l^{S_2}$ and $f_r^{S_2}$, respectively. The pictures show that S_2 satisfies pairs: $\langle 0, 2\rangle$ (a), $\langle 1, 2\rangle$ (b), and $\langle 2, 0\rangle$ (c).

between geometric curves.

We say that pair $\langle x_1, y_1\rangle$ *precedes wrt* x (*precedes wrt* y) a pair $\langle x_2, y_2\rangle$ when $x_1 \leq x_2$ ($y_1 \leq y_2$). We denote this relationship by $\preceq_x$ ($\preceq_y$). For example $\langle 3, 1\rangle \preceq_y \langle 3, 5\rangle$. We say that pair $\langle x_1, y_1\rangle$ *precedes* a pair $\langle x_2, y_2\rangle$ when $\langle x_1, y_1\rangle$ precedes $\langle x_2, y_2\rangle$ both wrt x and wrt y. We denote this relationship by $\preceq$. For example $\langle 3, 4\rangle \preceq \langle 3, 5\rangle$. Two pairs $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$ are *incomparable* if none of them precedes the other, i.e., if $\langle x_1, y_1\rangle \npreceq \langle x_2, y_2\rangle$ and $\langle x_2, y_2\rangle \npreceq \langle x_1, y_1\rangle$. The incomparability relationship is denoted by $\nsim$, as, for example, in $\langle 3, 4\rangle \nsim \langle 2, 5\rangle$. Based on the above definition, if μ_i satisfies $\langle x, y\rangle$, then it satisfies any pair $\langle w, z\rangle$ such that $\langle x, y\rangle \preceq \langle w, z\rangle$.

A set S of pairs of non-negative integers $\langle x, y\rangle$ is *succinct* if the pairs of S are pairwise incomparable. Given two sets S and S' of pairs of integers, S' *precedes* S if for any pair $p \in S$ there exists at least one pair $p' \in S'$ such that $p' \preceq p$. For example $\{\langle 0, 4\rangle\langle 3, 3\rangle\langle 5, 4\rangle\} \preceq \{\langle 0, 5\rangle\langle 4, 5\rangle\}$. Also, S' *reduces* S if $S' \preceq S$ and $S' \subseteq S$. Further, if S' is succinct and reduces S, S' is a *gist* of S. For example $\{\langle 0, 4\rangle\langle 3, 3\rangle\}$ is a gist of $\{\langle 0, 4\rangle\langle 3, 3\rangle\langle 5, 4\rangle\}$.

Let S be a set of non-negative integer pairs. The following properties hold.

Property 1 *The gist of S is unique and is the smallest set preceding S.*

Proof: Suppose, for a contradiction, that S has two gists S' and S'' with $S' \neq S''$. Without loss of generality, let p' be a pair such that $p' \in S'$ and $p' \notin S''$. From the definition of gist we have that $p' \in S$. Since $p' \notin S''$, there exists a pair $p'' \in S''$ such that $p'' \preceq p'$. We have that $p'' \in S$. We have a contradiction because S', which is a gist of S, contains the pair p' of S instead of the pair $p'' \preceq p'$. □

In the following, the unique gist of S is denoted by $\hat{S}$.

Property 2 *Let $p_1 = \langle x_1, y_1\rangle$ and $p_2 = \langle x_2, y_2\rangle$ be two elements of S. If S is succinct, then $p_1 \preceq_x p_2 \Leftrightarrow p_2 \preceq_y p_1$.*

Proof: Suppose for contradiction that $p_1 \preceq_x p_2$ and $p_1 \preceq_y p_2$. It follows that $p_1 \preceq p_2$. Hence, S is not a succinct set, contradicting the hypothesis. The opposite part of the proof is analogous. □

Property 3 *If S is succinct, then the relationship $\preceq_x$ induces a total order on S. Such a total order is an inverse total order with respect to relationship $\preceq_y$.*

Proof: The statement is a consequence of Property 2. There are no pairs in S with the same value of x or of y because ini that case they would be ordered wrt $\preceq$ relationship. □

Denote by $x^{\max}(S)$ ($y^{\max}(S)$) the maximum value of x_i (y_i) in any pair $\langle x_i, y_i\rangle \in S$.

Property 4 *If S is succinct, then $|S| \leq x^{\max}(S)$ and $|S| \leq y^{\max}(S)$.*

Proof: We have $0 \leq x \leq x^{\max}$. Since, by Property 3, the relationship $\preceq_x$ induces a total order on S, the statement follows. □

In the following we consider how unions and intersections can be efficiently computed on integer pairs sets when the values of the integers are suitably bounded.

Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. We assume that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to the $\preceq_x$ relationship. The set $S_i = S_1 \cap S_2$,

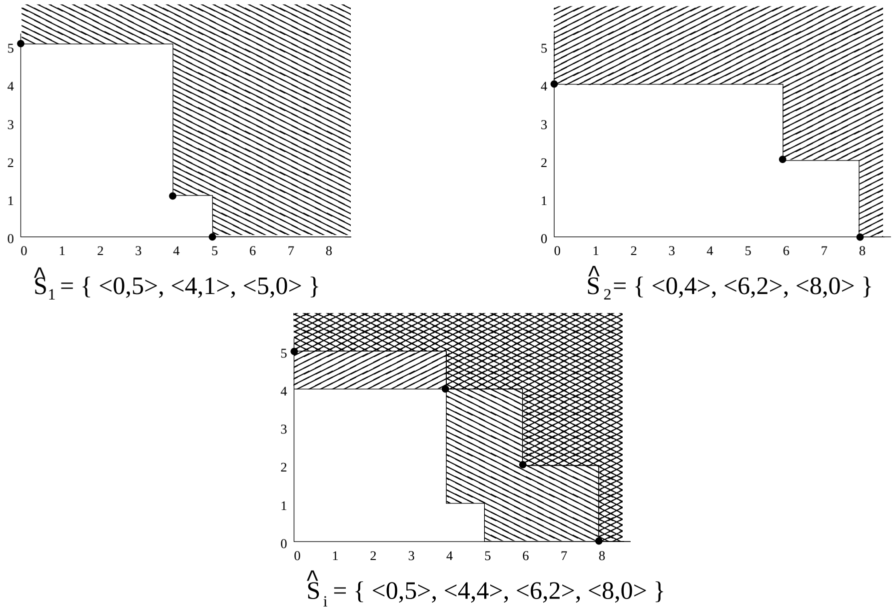


Figure 6: Computation of $\hat{S}_i$, given $\hat{S}_1$ and $\hat{S}_2$. An integer pair $\langle x, y \rangle$ belonging to a gist, is represented by a black circle with coordinates (x, y).

by definition, contains all the pairs that are contained both in S_1 and S_2. Since S_i is an infinite set, we compute its gist $\hat{S}_i$. Observe that $\hat{S}_i$ contains all the pairs preceded by at least one pair of both $\hat{S}_1$ and $\hat{S}_2$ and that are not preceded by any other pair of $\hat{S}_i$. Also, observe that, as shown by pair $\langle 4, 4 \rangle \in \hat{S}_i$ in Figure 6, $\hat{S}_i$ may contain some pair p such that $p \notin \hat{S}_1$ and $p \notin \hat{S}_2$.

Lemma 1 *Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists, sorted with respect to the $\preceq_x$ relationship, such that $\hat{S}_1 = \{\langle x_1, y_1 \rangle, \ldots, \langle x_j, y_j \rangle, \langle x_{j+1}, y_{j+1} \rangle, \ldots, \langle x_m, y_m \rangle\}$ and $\langle x, y \rangle \in \hat{S}_2$. If $x_j \leq x < x_{j+1}$ then $\langle x, max(y, y_j) \rangle \in S_i$, where $S_i = S_1 \cap S_2$.*

Proof: For each pair $\langle x_k, y_k \rangle \in \hat{S}_1$, with $k = 1, \ldots, j$, by $x_k \leq x$, there exist infinite pairs $\langle x, y_k + m \rangle \in S_1$, with $m \geq 0$. By Property 3, $\langle x, y_j \rangle \preceq \langle x, y_k + m \rangle$, for any $k = 1, \ldots, j$ and $m \geq 0$. Hence, by $x_j < x$, we have $\langle x_j, y_j \rangle \preceq \langle x, max(y, y_j) \rangle$. Since $\langle x, max(y, y_j) \rangle$ is preceded by $\langle x_j, y_j \rangle \in \hat{S}_1$ and by $\langle x, y \rangle \in \hat{S}_2$, the statement follows. □

The following algorithm, called Gist_Intersection, starting from the first pairs of the two set, recursively compares two pairs belonging to different sets, inserts in $\hat{S}_i$ the correct one and moves forward in at least one of the two sets in order to obtain the new pairs to compare. When one of the two sets is empty, all the pairs of the other set are inserted in $\hat{S}_i$. Namely, when comparing the two pairs $p_1 = \langle x_1, y_1 \rangle$ and $p_2 = \langle x_2, y_2 \rangle$, if $x_1 = x_2$ then insert $\langle x_1, \max(y_1, y_2) \rangle$ in $\hat{S}_i$ and move forward in both the sets. Analogously, if $y_1 = y_2$ then insert $\langle \max(x_1, x_2), y_1 \rangle$ in $\hat{S}_i$ and move forward in both the sets. If $x_1 \neq x_2$ and $y_1 \neq y_2$ then consider the pair with minimum x, say p_1, and insert in $\hat{S}_i$, by Lemma 1, the pair $\langle x_1, \max\{y_1, y'_2\} \rangle$, where $\langle x'_2, y'_2 \rangle \in \hat{S}_2$, $x'_2 < x_2$ and there is no pair $\langle x''_2, y''_2 \rangle \in \hat{S}_2$ such that $x'_2 < x''_2 < x_2$, if such a pair is not preceded by the last pair inserted. In this case move forward in $\hat{S}_1$.

Algorithm 1 GIST_INTERSECTION

Input: The gists $\hat{S}_1$ and $\hat{S}_2$ of two sets S_1 and S_2 of integer pairs, sorted with respect to the $\preceq_x$ relationship.

Output: The gist $\hat{S}_i$ of $S_i = S_1 \cap S_2$, sorted with respect to the $\preceq_x$ relationship.

```
p_1 = first(Ŝ_1);
p_2 = first(Ŝ_2);
Ŝ_i = ∅
INTERSECTION(p_1, p_2, Ŝ_1, Ŝ_2, Ŝ_i);
return Ŝ_i;

procedure INTERSECTION(pair_1 = ⟨x_1, y_1⟩, pair_2 = ⟨x_2, y_2⟩, A_1, A_2, A_i)
if pair_1 ≠ null & pair_2 ≠ null then
    if x_1 == x_2 then
        A_i.add(⟨x_1, max{y_1, y_2}⟩);
        pair_1 = A_1.next();
        pair_2 = A_2.next();
    else if y_1 == y_2 then
        A_i.add(⟨max{x_1, x_2}, y_1⟩);
        pair_1 = A_1.next();
        pair_2 = A_2.next();
    else if x_1 < x_2 then
        pair_temp = A_2.prev();
        if A_i.getLast() ⋠ ⟨x_1, max{y_1, y_temp}⟩ then
            A_i.add(⟨x_1, max{y_1, y_temp}⟩);
            pair_1 = A_1.next();
            pair_2 = A_2.next();
        end if
    else
        pair_temp = A_1.prev();
        if A_i.getLast() ⋠ ⟨x_2, max{y_2, y_temp}⟩ then
            A_i.add(⟨x_2, max{y_2, y_temp}⟩);
            pair_1 = A_1.next();
            pair_2 = A_2.next();
        end if
    end if
    INTERSECTION(pair_1, pair_2, A_1, A_2, A_i);
else
    while pair_2 ≠ null do
        A_i.add(pair_2);
        pair_2 = A_2.next();
    end while
    while pair_1 ≠ null do
        A_i.add(pair_1);
        pair_1 = A_1.next();
    end while
end if
```

Lemma 2 (Intersection complexity by size) *Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. Suppose that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to the $\preceq_x$ relationship. The gist $\hat{S}_i$ of $S_i = S_1 \cap S_2$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(|\hat{S}_1| + |\hat{S}_2|)$ time.*

Proof: Apply Algorithm GIST_INTERSECTION. At every step the algorithm moves forward in at least one of the two sets. Hence, there are $O(|\hat{S}_1| + |\hat{S}_2|)$ operations to execute. Since each operation can be executed in $O(1)$ time, the statement follows. □

Lemma 3 (Intersection complexity by value) *Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. Suppose that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to the $\preceq_x$ relationship. The gist $\hat{S}_i$ of $S_i = S_1 \cap S_2$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(\min(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2))$ time. Further, $x^{\max}(\hat{S}_i)$ is $\max(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2))$.*

Proof: Apply Algorithm GIST_INTERSECTION. At every step the algorithm moves forward in the set whose currently considered pair has smallest x. Hence, each pair is overcome after a number of steps that is proportional to the value of its x. Hence, after $O(\min(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2))$ steps the set $\hat{S}_j$, with $j \in \{1, 2\}$, with minimum value of $x^{\max}(\hat{S}_j)$ becomes empty. Assuming that, using an appropriate data structure, all the remaining pairs of the non-empty set can be inserted in $\hat{S}_i$ in $O(1)$ time, the time complexity bound follows. The second part of the statement follows from the fact that the last pair of $\hat{S}_i$ is the last pair of the non-empty set, which is the set $\hat{S}_j$, with $j \in \{1, 2\}$, with maximum value of $x^{\max}(\hat{S}_j)$. □

Lemma 4 *Let S_j, $j = 1, \ldots, k$, be k sets of integer pairs and let $\hat{S}_j$ be their gists, each one sorted with respect to the $\preceq_x$ relationship. The gist of $S_1 \cap S_2 \cap \cdots \cap S_k$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(\sum_{j=1}^{k}(x^{\max}(\hat{S}_j)))$ or, equivalently, in $O(\sum_{j=1}^{k}(|\hat{S}_j|))$ time.*

Proof: Let I_j be the intersection of the first j sets, that is, $I_j = S_1 \cap S_2 \cap \cdots \cap S_j$, and let $\hat{I}_j$ be its gist. Observe that $I_1 = S_1$. In order to obtain $\hat{I}_k$, we repeatedly apply Lemma 3 to compute each $\hat{I}_j$, with $j = 2, \ldots, k$, starting from $\hat{I}_{j-1}$ and $\hat{S}_j$. It is easy to prove by induction that $x^{\max}(\hat{I}_j) = O(\max\limits_{m=1,\ldots,j}(x^{\max}(\hat{S}_m)))$. By Lemma 3 it follows that the computation of $\hat{I}_j$ starting from $\hat{I}_{j-1}$ and $\hat{S}_j$ can be performed in $O(\min(x^{\max}(\hat{I}_{j-1}), x^{\max}(\hat{S}_j))) = O(\min(\max\limits_{m=1,\ldots,j-1}(x^{\max}(\hat{S}_m)), x^{\max}(\hat{S}_j)))$ time. Therefore, denoted by $x^{\max}_* = \max\limits_{j=1,\ldots,k}(x^{\max}(\hat{S}_j))$, we have that $\hat{I}_k$ can be computed in time

$$\sum_j O(\min(\max_{m=1,\ldots,j-1}(x^{\max}(\hat{S}_m), x^{\max}(\hat{S}_j))) \leq O(\sum_j \min(x^{\max}_*, x^{\max}(\hat{S}_j))) =$$

$$= O(\sum_j x^{\max}(\hat{S}_j)).$$

Since $|\hat{S}_j|$ is $O(x^{\max}(\hat{S}_j))$, we equivalently have that $\hat{I}_k$ can be computed in $O(\sum_j(|\hat{S}_j|))$ time. □

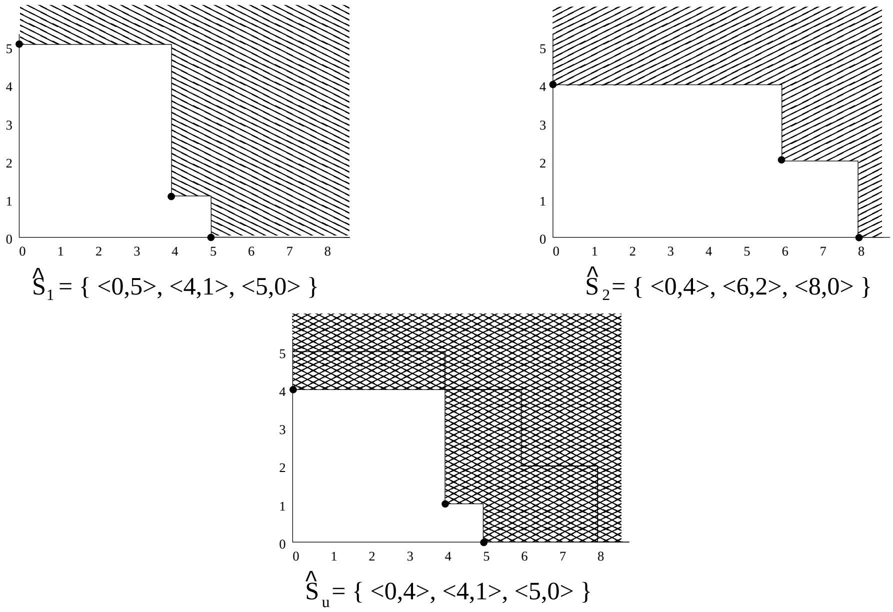


Figure 7: Computation of $\hat{S}_u$, given $\hat{S}_1$ and $\hat{S}_2$. An integer pair $\langle x, y\rangle$ belonging to a gist, is represented by a black circle with coordinates (x, y).

Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. We assume that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to the $\preceq_x$ relationship. The set $S_u = S_1 \cup S_2$, by definition, contains all the pairs that are contained either in S_1 or in S_2. Since S_u is an infinite set, we compute its gist $\hat{S}_u$. Observe that $\hat{S}_u$ contains all the pairs preceded by at least one pair of either $\hat{S}_1$ or $\hat{S}_2$ and that are not preceded by any other pair of $\hat{S}_u$. Also, observe that if a pair $p \in \hat{S}_u$, then either $p \in \hat{S}_1$ or $p \in \hat{S}_2$.

The following algorithm, called GIST_UNION, starting from the first pairs of the two set, recursively compares two pairs, inserts in $\hat{S}_u$ the correct one and moves forward in at least one of the two sets in order to obtain the new pairs to compare. When one of the two sets is empty all the pairs of the other set are inserted in $\hat{S}_u$. Namely, when comparing the two pairs $p_1 = \langle x_1, y_1\rangle$ and $p_2 = \langle x_2, y_2\rangle$, if $p_1 \sim p_2$ then consider the pair with minimum x, say p_1, insert it in $\hat{S}_u$ and move forward in $\hat{S}_1$. If $p_1 \preceq p_2$ then move forward in $\hat{S}_2$ without inserting any pair in $\hat{S}_u$. The same if $p_2 \preceq p_1$. If $p_1 \preceq p_2$ and $p_2 \preceq p_1$, that is, $p_1 = p_2$, then insert either p_1 or p_2 in $\hat{S}_u$ and move forward in both the sets.

Lemma 5 (Union complexity by size) *Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. Suppose that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to the $\preceq_x$ relationship. The gist $\hat{S}_u$ of $S_u = S_1 \cup S_2$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(|\hat{S}_1| + |\hat{S}_2|)$ time.*

Proof: Apply Algorithm GIST_UNION. At every step the algorithm moves forward in at least one of the two sets, hence there are $O(|\hat{S}_1| + |\hat{S}_2|)$ operations to execute. Since each operation can be executed in $O(1)$ time, the statement follows. □

Lemma 6 (Union complexity by value) *Let S_1 and S_2 be two sets of integer pairs and let $\hat{S}_1$ and $\hat{S}_2$ be their gists. Suppose that $\hat{S}_1$ and $\hat{S}_2$ are sorted with respect to*

Algorithm 2 GIST_UNION

Input: The gists $\hat{S}_1$ and $\hat{S}_2$ of two sets S_1 and S_2 of integer pairs, sorted with respect to the $\preceq_x$ relationship.

Output: The gist $\hat{S}_u$ of $S_u = S_1 \cup S_2$, sorted with respect to the $\preceq_x$ relationship.

$p_1 = \text{first}(\hat{S}_1)$;
$p_2 = \text{first}(\hat{S}_2)$;
$\hat{S}_u = \emptyset$;
UNION$(p_1, p_2, \hat{S}_1, \hat{S}_2, \hat{S}_u)$;
return $\hat{S}_u$;

procedure UNION$(pair_1 = (x_1, y_1),\ pair_2 = (x_2, y_2),\ A_1,\ A_2, A_u)$
if $pair_1 \neq$ null & $pair_2 \neq$ null **then**
    if $x_1 > x_2$ & $y_1 < y_2$ **then**
        A_u.add($pair_2$);
        $pair_2 = A_2$.next();
    end if
    if $x_1 < x_2$ & $y_1 > y_2$ **then**
        A_u.add($pair_1$);
        $pair_1 = A_1$.next();
    end if
    if $pair_1 == pair_2$ **then**
        A_u.add($pair_1$);
    end if
    if $pair_1 \preceq pair_2$ **then**
        $pair_2 = A_2$.next();
    end if
    if $pair_2 \preceq pair_1$ **then**
        $pair_1 = A_1$.next();
    end if
    UNION$(pair_1, pair_2, A_1, A_2, A_u)$;
else
    while $pair_2 \neq$ null **do**
        A_u.add($pair_2$);
        $pair_2 = A_2$.next();
    end while
    while $pair_1 \neq$ null **do**
        A_u.add($pair_1$);
        $pair_1 = A_1$.next();
    end while
end if

the $\preceq_x$ relationship. The gist $\hat{S}_u$ of $S_u = S_1 \cup S_2$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(\max(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2))$ time. Further, $x^{\max}(\hat{S}_u)$ is $\min(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2))$.

Proof: Apply Algorithm Gist_Union. The time complexity bound follows from Lemma 5 and from the fact that, by Property 4, we have $|\hat{S}_1| \leq x^{\max}(\hat{S}_1)$ and $|\hat{S}_2| \leq y^{\max}(\hat{S}_2)$. Consider the last pair of $\hat{S}_u$, that is, by Property 3, the pair $\langle x, y\rangle$ such that $x = x^{\max}(\hat{S}_u)$ and $y = 0$. By construction, such a pair is either $p_1 = \langle x^{\max}(\hat{S}_1), 0\rangle \in \hat{S}_1$ or $p_2 = \langle x^{\max}(\hat{S}_2), 0\rangle \in \hat{S}_2$. The second part of the statement follows from the fact that pair $\langle \min(x^{\max}(\hat{S}_1), x^{\max}(\hat{S}_2)), 0\rangle$ precedes both p_1 and p_2. □

Analogously to Lemma 4, the following lemma shows that the gist of the union of k sets of integer pairs can be computed in time linear in the sum of the sizes of their gists.

Lemma 7 *Let S_j, $j = 1, \ldots, k$, be k sets of integer pairs and let $\hat{S}_j$ be their gists, each one sorted with respect to the $\preceq_x$ relationship. The gist of $S_u = S_1 \cup S_2 \cup \cdots \cup S_k$, sorted with respect to the $\preceq_x$ relationship, can be computed in $O(\sum_{j=1}^{k}(x^{\max}(\hat{S}_j)))$ or, equivalently, in $O(\sum_{j=1}^{k}(|\hat{S}_j|))$ time.*

Proof: We prove the statement by induction on the number of sets k. By Lemma 6 the statement is true for $k = 2$. Suppose the statement is true for $k - 1$. We prove the statement for k. Let $U_h = S_1 \cup S_2 \cup \cdots \cup S_h$ be the union of h sets. Starting from the inductive hypothesis that $\hat{U}_{k-1}$ can be computed in $O(\sum_{j=1}^{k-1}(x^{\max}(\hat{S}_j)))$, we prove that $\hat{U}_k$ can be computed in $O(\sum_{j=1}^{k}(x^{\max}(\hat{S}_j)))$. Consider the set S_m such that $x^{\max}(S_m) \geq x^{\max}(S_j)$, for $j = 1, \ldots, k$. Let $U_{k-1} = \bigcup_{j=1,\, j\neq m}^{k} S_j$. Compute $\hat{U}_{k-1}$ in $O(\sum_{j=1}^{k-1}(x^{\max}(\hat{S}_j)))$ time. By Lemma 6, we have that, starting from $\hat{U}_{k-1}$, the gist of $U_k = U_{k-1} \cup S_m$ can be computed in $O(\max(x^{\max}(\hat{U}_{k-1}), x^{\max}(\hat{S}_m)) = O(x^{\max}(\hat{S}_m))$ time. Hence, we obtain $\hat{U}_k$ in $O(\sum_{j=1}^{k-1}(x^{\max}(\hat{S}_j)) + x^{\max}(\hat{S}_m)) = O(\sum_{j=1}^{k}(x^{\max}(\hat{S}_j)))$ time. By Property 4 we have also that the computation can be performed in $O(\sum_{j=1}^{k}(|\hat{S}_j|))$ time. □

The above investigation about integer-pair sets allows us to represent in a finite way the infinite admissible set of a component, which is used to describe the depths of all the embeddings of the pertinent graph of the component. Namely, the infinite set $A(\mu_i)$ of integer pairs satisfied by component μ_i will be represented by its gist $\hat{A}(\mu_i)$, maintained ordered wrt the $\preceq_x$ relationship. Also, the intersections and unions of such admissible sets can be performed efficiently by using the above described Gist_Intersection and Gist_Union algorithms.

Further, with respect to the gist of a generic integer-pair set, the gist of an admissible set satisfies additional properties, described in the following, that can be used to simplify the operations and to bound the complexity of the algorithms. Let μ_i be a component, G_{μ_i} its pertinent graph and n_i the number of vertices of G_{μ_i}.

Property 5 *If $\langle x, y\rangle \in \hat{A}(\mu_i)$, then $\langle y, x\rangle \in \hat{A}(\mu_i)$.*

Proof: Suppose, for contradiction, that $\langle x, y\rangle \in \hat{A}(\mu_i)$ and $\langle y, x\rangle \notin \hat{A}(\mu_i)$. Since $\langle x, y\rangle \in A(\mu_i)$, as stated above, we have $\langle y, x\rangle \in A(\mu_i)$. Hence, there exists a pair $\langle x', y'\rangle \in A(\mu_i)$ such that $\langle x', y'\rangle \preceq \langle y, x\rangle$. Hence, $\langle y', x'\rangle \in A(\mu_i)$. We have, by construction, $\langle y', x'\rangle \preceq \langle x, y\rangle$, contradicting the hypothesis $\langle x, y\rangle \in \hat{A}(\mu_i)$. □

Property 6 $\hat{A}(\mu_i)$ *contains exactly one pair* $\langle x, y\rangle$ *with* $x = 0$ *and one pair* $\langle x', y'\rangle$ *with* $y' = 0$. *In such pairs* $y = x' = O(n_i)$.

Proof: Let F be the set of the internal faces of an arbitrary embedding $\Gamma_{G_{\mu_i}}$ of G_{μ_i}. Consider the trivial partition of F into two sets such that one is empty and the other one coincides with F. Such a partition implies that $A(\mu_i)$ contains a pair $\langle x^*, y^*\rangle$ with $x^* = 0$ and $y^* \leq |F|$. Since any pair $\langle x, y\rangle$ preceding $\langle x^*, y^*\rangle$ has $x = 0$, $\hat{A}(\mu_i)$ contains at least one pair $\langle x, y\rangle$ with $x = 0$. Two such pairs can not be contained into $\hat{A}(\mu_i)$ since $\hat{A}(\mu_i)$ is succinct. The bound on the value of y is due to the fact that $y \leq y^* \leq |F|$ and that $|F| = O(n_i)$. Analogous considerations show that there exists exactly one pair $\langle x', y'\rangle \in \hat{A}(\mu_i)$ with $x' = O(n_i)$ and $y' = 0$. By Property 5 we have $y = x'$. □

Property 7 $\hat{A}(\mu_i)$ *is finite and* $|\hat{A}(\mu_i)|$ *is* $O(n_i)$.

Proof: By Property 6 $\hat{A}(\mu_i)$ contains two pairs $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$, with $x_1 = 0$ and $x_2 = O(n_i)$, that represent the first and the last element of $\hat{A}(\mu_i)$. Since, by Property 3, $\hat{A}(\mu_i)$ is totally ordered wrt the $\preceq_x$ relationship, it follows that $|\hat{A}(\mu_i)| = O(n_i)$. □

4 Computing a Minimum-Depth Embedding of a Biconnected Planar Graph

The minimum-depth embedding of a biconnected planar graph G can be found by applying for each edge e of G the algorithm presented in this section, which computes the minimum-depth embedding of G with edge e on the external face.

Such a computation is performed by means of two traversals of the SPQR-tree $\mathcal{T}$ of G, rooted at e.

The first traversal is a bottom-up traversal. Its purpose is to label each virtual edge e_i, corresponding to node μ_i, with suitable values in order to describe the properties, with respect to the depth, of all possible embeddings of the pertinent graph G_{μ_i}.

Such values are:

- The gist $\hat{A}(\mu_i)$ of the admissible set of μ_i
- The distance between $f_l^{\mu_i}$ and $f_r^{\mu_i}$ in the (u_i, v_i)-dual of G_{μ_i}, which is called the *thickness* of μ_i and is denoted by $t(\mu_i)$.

In [3], where the concept of thickness was also used, it is shown that $t(\mu_i)$ is independent on the embedding of the pertinent graph G_{μ_i} of μ_i. Hence, $t(\mu_i)$ can also be defined as the distance between $f_l^{\mu_i}$ and $f_r^{\mu_i}$ in the (u_i, v_i)-dual of $sk(\mu_i)$, where each edge of the (u_i, v_i)-dual of $sk(\mu_i)$, corresponding to a virtual edge of $sk(\mu_i)$ which represents a child component ν, is associated with a weight that is the thickness of ν.

At the end of the bottom-up traversal of the SPQR-tree $\mathcal{T}$, the unique child component of the root e of $\mathcal{T}$ is labeled with the gist of the admissible set of G, describing all possible depths of the embeddings of G with e on the external face. From such an admissible set an optimal pair can be selected and used in a top-down traversal of $\mathcal{T}$ to select a suitable embedding for the skeleton of each node of $\mathcal{T}$.

4.1 Labeling an SPQR-tree with Minimum-Depth Embedding Descriptors

During the bottom-up traversal of $\mathcal{T}$, for each component μ, we compute its thickness $t(\mu)$ and its gist $\hat{A}(\mu)$ based on the analogous values computed for its children.

First, we describe the general strategy to compute $\hat{A}(\mu)$ depending on the fact that μ is a series, a rigid, or a parallel node (the computation of $t(\mu)$ is easier and, sometimes, trivial). Series and rigid cases, addressed in detail in the first two subsections of this section, have several analogies, while the parallel case, addressed in the last subsection, is much more complex and needs a more sophisticated approach.

For the series and rigid cases, our strategy is based on the fact that the set of all embeddings of the pertinent graph G_μ of μ can be suitably partitioned, and each block of the partition can be separately analyzed. For the parallel case, instead, our strategy is based on the exploration of a bounded-size subset of the set of all possible embeddings of G_μ which maintains the same admissible set.

Let μ be a node of $\mathcal{T}$ and let $sk(\mu)$ be its skeleton. If μ is a series $sk(\mu)$ has a unique embedding Γ^1_μ, if μ is a rigid $sk(\mu)$ admits two embeddings Γ^1_μ and Γ^2_μ, and if μ is a parallel with k child components $sk(\mu)$ admits $k!$ embeddings Γ^h_μ, with $h = 1, \ldots, k!$. Each embedding of G_μ is compatible with exactly one embedding Γ^j_μ of $sk(\mu)$. Hence, the embeddings of $sk(\mu)$ induce a partition on the embeddings of G_μ. For series and rigid nodes, in order to compute $A(\mu)$ through all possible embeddings of G_μ, we first compute the admissible sets $A^j(\mu)$, restricted to those embeddings of G_μ corresponding to a single embedding Γ^j_μ of $sk(\mu)$, and then perform their union.

Given an embedding Γ_{G_μ} of the pertinent graph G_μ of μ we distinguish two types of faces. We call *children faces* the faces of Γ_{G_μ} that are also faces of some Γ_{G_ν}, with ν child of μ. We call *skeleton faces* all the other faces.

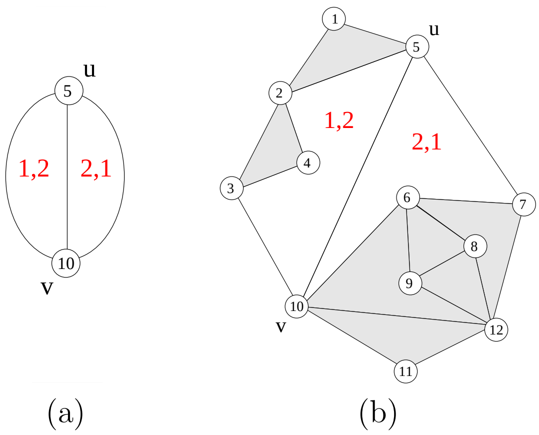


Figure 8: (a) Embedding $\Gamma^1_{P_1}$ of the skeleton of parallel component P_1. (b) An embedding Γ_{P_1} of P_1 compatible with $\Gamma^1_{P_1}$. Skeleton faces are drawn white and children faces are drawn grey.

Essentially, "shrinking" each pertinent graph of the children of μ into a single (virtual) edge, we obtain that the skeleton faces of Γ_{G_μ} transform into the faces of an embedding Γ^j_μ of $sk(\mu)$.

Observe that, once the embedding Γ_μ^j of $sk(\mu)$ has been fixed, the distances of each skeleton face of any embedding of Γ_{G_μ} from f_l^μ and f_r^μ depend on the values $t(\nu_1), \ldots, t(\nu_k)$ only, which, in turn, are independent on the embedding of the child components of μ. Hence, each face f of Γ_μ^j can be labeled with its depths, denoted by $d_l(f)$ and $d_r(f)$, which correspond to the above distances. Fig. 8.a shows an embedding $\Gamma_{P_1}^1$ of the skeleton of parallel component P_1 and Fig. 8.b shows an embedding Γ_{P_1} of the pertinent graph of P_1 compatible with $\Gamma_{P_1}^1$. Notice that skeleton faces, which are drawn white, have the same values of depth in $\Gamma_{P_1}^1$ and in Γ_{P_1}.

For $sk(\mu)$ definitions analogous to those given for G_μ can be given. In particular, we say that Γ_μ^j *satisfies* the pair of non-negative integers $\langle x, y\rangle$ if it is possible to find a partition of its internal faces into two sets, denoted by F_l and F_r, such that each face $f \in F_l$ has $d_l(f) \leq x$ and each face $f \in F_r$ has $d_r(f) \leq y$. The infinite set of integer pairs satisfied by Γ_μ^j is the *admissible set of* Γ_μ^j, and is denoted by $A(\Gamma_\mu^j)$.

Once the embedding Γ_μ^j of $sk(\mu)$ has been fixed, the admissible set $A^j(\mu)$, i.e., the admissible set of μ restricted to the embeddings compatible with Γ_μ^j, can be computed starting from $A(\Gamma_\mu^j)$, from the admissible set $A(\nu_i)$ of each child component ν_i, with $i = 1, \ldots, \delta(\mu)$, and from the depths of its left and right external faces $f_l^{\nu_i}$ and $f_r^{\nu_i}$.

Namely, μ satisfies the integer pair $\langle x, y\rangle$ if $\langle x, y\rangle \in A(\Gamma_\mu^j)$ and each child component ν_i satisfies a pair $\langle x^i, y^i\rangle$ such that:

- $x^i + d_l(f_l^{\nu_i}) \leq x$ or $x^i + d_r(f_l^{\nu_i}) \leq y$, and
- $y^i + d_r(f_r^{\nu_i}) \leq y$ or $y^i + d_l(f_r^{\nu_i}) \leq x$.

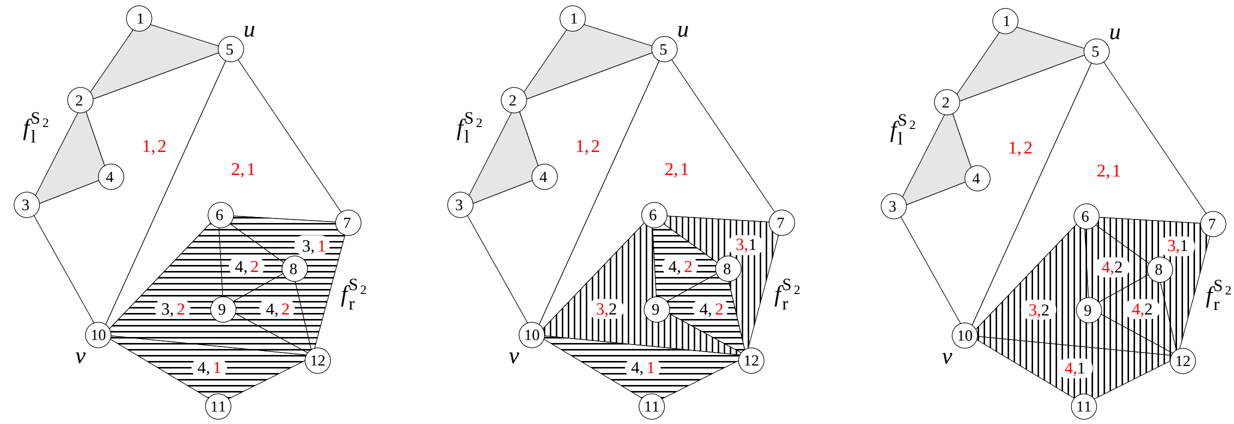
Hence, in order to obtain $A^j(\mu)$, we compute for each child component ν_i of μ the set of integer pairs that are satisfied by ν_i when inserted into Γ_μ^j, that is, the set of integer pairs that verify the conditions above.

Namely, let μ be a node of the SPQR-tree $\mathcal{T}$, let Γ_μ^j be an embedding of $sk(\mu)$, let ν be a child of μ, and let $\langle x, y\rangle$ be a pair of non-negative integers. Node ν *satisfies* $\langle x, y\rangle$, *nested into* Γ_μ^j, if the pertinent graph G_μ of μ admits an embedding Γ_{G_μ}, compatible with Γ_μ^j, where it is possible to find a partition of the set of the children faces corresponding to the internal faces of ν into two sets, denoted by F_l and F_r, such that all faces in F_l have distance from f_l^μ less or equal than x and all faces in F_r have distance from f_r^μ less or equal than y. In Fig. 9 it is shown how component S_2 satisfies, nested into $\Gamma^j(P_1)$, pairs $\langle 0, 2\rangle$, $\langle 3, 2\rangle$, and $\langle 4, 0\rangle$, with the corresponding partitions of its internal faces. Each internal face of S_2 is labeled with a pair of integers representing its distance from left and right external faces of $\Gamma^j(P_1)$, respectively.

The infinite set of integer pairs satisfied by component ν, nested into Γ_μ^j, is the *admissible set of* ν *into* Γ_μ^j, and is denoted by $A(\nu|\Gamma_\mu^j)$. The gist of $A(\nu|\Gamma_\mu^j)$ is denoted by $\hat{A}(\nu|\Gamma_\mu^j)$ and assumed ordered with respect to the $\preceq_x$ relationship.

Lemma 8 *Given an embedding Γ_μ^j of $sk(\mu)$, the admissible set $A^j(\mu)$ of G_μ (restricted to those embeddings of G_μ corresponding to Γ_μ^j) can be obtained by intersecting the $\delta(\mu)$ sets $A(\nu_i|\Gamma_\mu^j)$ and $A(\Gamma_\mu^j)$.*

Proof: The proof is based on the fact that the distances between the internal faces of the embedding of a component ν_k and the external faces f_l^μ and f_r^μ of μ are independent on the embedding of other child components of μ.



(a) (b)

Figure 9: An embedding of component S_2, nested into $\Gamma^j(P_1)$. Figures (a), (b), and (c) represent three possible partitions of the internal faces of S_2. Faces in F_l are filled with vertical lines, faces in F_r are filled with horizontal lines, skeleton faces are drawn white, and children faces not belonging to S_2 are drawn grey. Each face is labeled with two integers representing its distance from $f_l^{P_1}$ and $f_r^{P_1}$, respectively. The pictures show that S_2 satisfies, nested into $\Gamma^j(P_1)$, pairs: $\langle 0, 2\rangle$ (a), $\langle 3, 2\rangle$ (b), and $\langle 4, 0\rangle$ (c).

We first show that, given Γ^j_μ of $sk(\mu)$, a pair belonging to $A^j(\mu)$ of G_μ also belongs to $A(\nu_i|\Gamma^j_\mu)$ and $A(\Gamma^j_\mu)$. Second, we show that if a pair belongs to $A(\nu_i|\Gamma^j_\mu)$ and $A(\Gamma^j_\mu)$ then it belongs to $A^j(\mu)$ of G_μ.

Let $p = \langle x, y\rangle$ be a pair of non-negative integers belonging to $A^j(\mu)$ of G_μ, and let $A(\nu_k|\Gamma^j_\mu)$ be the admissible set of component ν_k nested into Γ^j_μ. Since $p \in A^j(\mu)$, there exists an embedding Γ_{G_μ} of G_μ, coherent with the embedding Γ^j_μ of $sk(\mu)$, whose internal faces can be partitioned into two sets F_l and F_r such that faces in F_l are at distance less or equal than x from f_l^μ and faces in F_r are at distance less or equal than y from f_r^μ. In order to show that p belongs to $A(\nu_k|\Gamma^j_\mu)$, it suffices to observe that the faces of the embedding Γ_{G_μ} that also belong to ν_k, can be partitioned into two sets $F'_l \subseteq F_l$ and $F'_r \subseteq F_r$ such that faces in F'_l are at distance less or equal than x from f_l^μ and faces in F'_r are at distance less or equal than y from f_r^μ. Analogously, it can be shown that p belongs to $A(\Gamma^j_\mu)$, since from F_l and F_r a suitable partition of the skeleton faces can be found as required by the definition of $A(\Gamma^j_\mu)$.

Conversely, let $p = \langle x, y\rangle$ be a pair of non-negative integers belonging to $A(\nu_i|\Gamma^j_\mu)$ and $A(\Gamma^j_\mu)$. An embedding Γ_{G_μ} of G_μ can be obtained from the embeddings of ν_i and $sk(\mu)$ that satisfy p. □

As said above, the admissible set $A(\mu)$ can be easily obtained as the union of the admissible sets $A^j(\mu)$ computed for any embedding Γ^j_μ of $sk(\mu)$.

4.1.1 The Series Case.

In the series case, as shown in Fig. 10, $sk(\mu)$ has exactly one embedding and such an embedding has no internal face. Hence, in order to compute $\hat{A}(\mu)$, it is not necessary to compute $A(\Gamma^1_\mu)$ and it is sufficient, by Lemma 8, to intersect the gists $\hat{A}(\nu_i|\Gamma^1_\mu)$ of the admissible sets of the child components ν_i nested into Γ^j_μ.

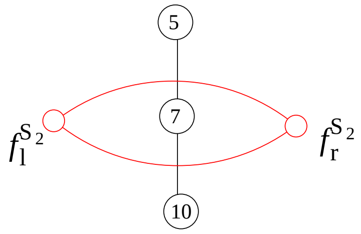


Figure 10: The unique embedding of $sk(S_2)$ and its (u,v)-dual.

We propose an algorithm, called NESTED_SERIES, which, given an S-node μ and one of its children ν, suitably builds a set S starting from $\hat{A}(\nu)$ and $t(\mu)$, and we show that $S = \hat{A}(\nu|\Gamma^1_\mu)$. The algorithm starts initializing S with $\hat{A}(\nu)$. Observe that, by Property 6, $\hat{A}(\nu)$ contains the two pairs $p_{first} = \langle 0, y_{max}\rangle$ and $p_{last} = \langle x_{max}, 0\rangle$, with $y_{max} = x_{max}$. For each pair $p^k = \langle x_k, y_k\rangle$ of $\hat{A}(\nu)$ define $p^k_{first} = \langle 0, max(y_k, x_k + t(\mu))\rangle$. Denote by $\overline{p}_{first}$ the p^k_{first} with minimum y and by $\overline{p}_{last}$ the pair obtained from $\overline{p}_{first}$ swapping the two elements x and y. If $\overline{p}_{first} \preceq p_{first}$ insert $\overline{p}_{first}$ into S and remove from S any pair p^* such that $\overline{p}_{first} \preceq p^*$. If $\overline{p}_{last} \preceq p_{last}$ append $\overline{p}_{last}$ to S and remove from S any pair p^* such that $\overline{p}_{last} \preceq p^*$.

Let μ be an S-node with children ν_i, for $i = 1, \ldots, \delta(\mu)$ and let $n(\nu_i)$ be the number of vertices of ν_i.

Lemma 9 *The thickness $t(\mu) = \min_i(t(\nu_i))$, for $i = 1, \ldots, \delta(\mu)$, can be computed in $O(\delta(\mu))$ time.*

Proof: As shown in Fig. 10, the (u,v)-dual graph of $sk(\mu)$ is made up of two vertices, corresponding to f^μ_l and f^μ_r, and $O(\delta(\mu))$ edges connecting them, each one corresponding to a child component ν_i and associated with a weight that is $t(\nu_i)$. Since, by definition, $t(\mu)$ is the distance between f^μ_l and f^μ_r in such a (u,v)-dual graph, the statement follows. □

Property 8 *The set S computed by Algorithm* NESTED_SERIES *is succinct.*

Proof: S is initialized with $\hat{A}(\nu_i)$, that is succinct, and, when the two pairs $\overline{p}_{first}$ and $\overline{p}_{last}$ are eventually added, all the pairs p^* preceded by them are removed. □

Property 9 *The set S computed by Algorithm* NESTED_SERIES *is a subset of $A(\nu|\Gamma^1_\mu)$.*

Proof: Consider a pair $p = \langle x, y\rangle \in S$. Two are the cases: either p was in $\hat{A}(\nu_i)$ or not. If $p \in \hat{A}(\nu_i)$ then, since μ is an S-node, $p \in A(\nu|\Gamma^1_\mu)$. If $p \notin \hat{A}(\nu)$, then p is either $\overline{p}_{first}$ or $\overline{p}_{last}$. Consider $p = \overline{p}_{first}$. We show that $p \in A(\nu|\Gamma^1_\mu)$ as follows. Consider the pair $p^k = \langle x_k, y_k\rangle$ such that $p^k_{first} = \overline{p}_{first}$. Since $p^k \in \hat{A}(\nu)$, there exists an embedding Γ^k_ν such that the set of the internal faces of Γ^k_ν can be partitioned into the two sets F^k_l and F^k_r such that all faces in F^k_l have distance from f^ν_l less or equal than x_k and all faces in F^k_r have distance from f^ν_r less or equal than y_k. For example, consider the partition of the internal faces of the embedding of component R_1 in Fig. 11.a which satisfies pair $\langle 1, 2\rangle$. Consider

Algorithm 3 NESTED_SERIES

Input: An S-node μ, with its thickness $t(\mu)$, and one of its children ν, with its gist $\hat{A}(\nu)$.
Output: The gist $\hat{A}(\nu|\Gamma^1_\mu)$ of the admissible set of ν nested into Γ^j_μ.

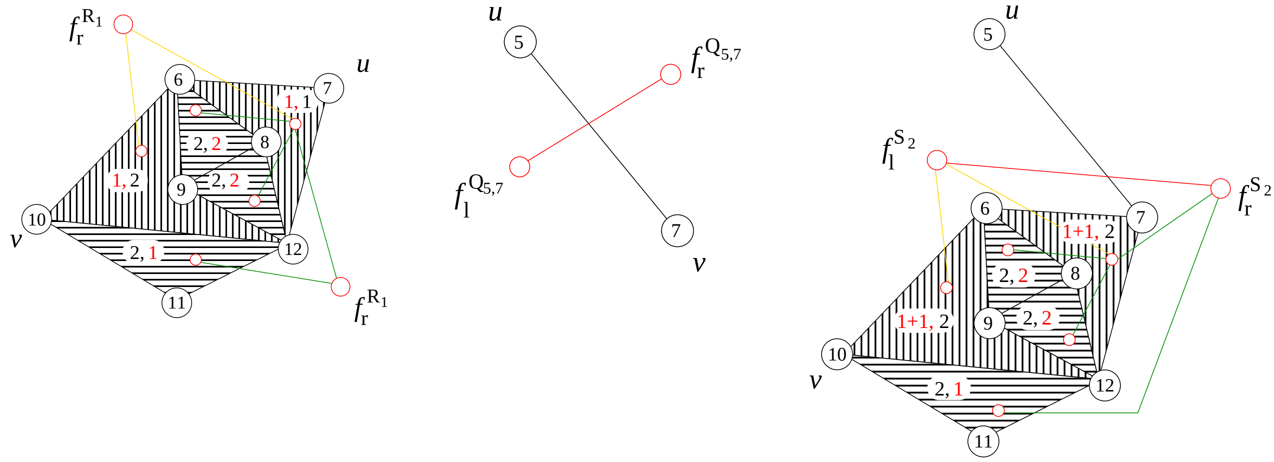
$S = \hat{A}(\nu)$;
$p_{first} = \hat{A}(\nu)$.getFirst();
for all $p^k = \langle x_k, y_k \rangle \in \hat{A}(\nu)$ **do**
  $p^k_{first} = \langle 0, \max\{y_k, x_k + t(\mu)\}\rangle$
  if $p^k_{first} \preceq \overline{p}_{first}$ **then**
    $\overline{p}_{first} = p^k_{first}$;
  end if
end for
if $\overline{p}_{first} \preceq p_{first}$ **then**
  S.addFirst($\overline{p}_{first}$);
  $\overline{p}_{last} = \overline{p}_{first}$.swapElements();
  S.addLast($\overline{p}_{last}$);
  for all $p^* \neq \overline{p}_{first} \in S$ and $p^* \neq \overline{p}_{last} \in S$ **do**
    if $\overline{p}_{first} \preceq p^*$ or $\overline{p}_{last} \preceq p^*$ **then**
      S.remove(p^*);
    end if
  end for
end if
return S;

any embedding $\Gamma^*_{G_\mu}$ such that $\Gamma^*_{G_\mu}$ restricted to ν is Γ^k_ν. As shown in Fig. 11.c, pair $p^k_{first} = \langle 0, max(y_k, x_k + t(\mu))\rangle$ belongs to $A(\nu|\Gamma^1_\mu)$, since each face of $\Gamma^*_{G_\mu}$ internal to ν_i is at distance less or equal than $max(y_k, x_k + t(\mu))$ from f^μ_r in G_μ. In fact, faces in F^k_l are at distance less or equal than x_k from f^ν_l which, in turn, is at distance $t(\mu))$ from f^μ_r, and faces in F^k_r are at distance less or equal than y_k from f^μ_r. Since $\overline{p}_{first} = \langle \overline{x}, \overline{y}\rangle \in A(\nu|\Gamma^1_\mu)$, we have that $\overline{p}_{last} = \langle \overline{y}, \overline{x}\rangle \in A(\nu|\Gamma^1_\mu)$. □

Property 10 *The set S computed by Algorithm* NESTED_SERIES *precedes $A(\nu|\Gamma^1_\mu)$.*

Proof: Since $\hat{A}(\nu|\Gamma^1_\mu) \preceq A(\nu|\Gamma^1_\mu)$, it is sufficient to show that $S \preceq \hat{A}(\nu|\Gamma^1_\mu)$. Suppose for contradiction that there exists a pair $p = \langle x, y\rangle$ such that $p \in \hat{A}(\nu|\Gamma^1_\mu)$ and there is not a pair $p' \in S$ such that $p' \preceq p$. We have $p \in \hat{A}(\nu|\Gamma^1_\mu)$. Hence, by definition, there exists an embedding Γ_{G_μ} such that all faces of ν can be partitioned into two sets F_l and F_r such that all faces in F_l are at distance less or equal than x from f^μ_l and all faces in F_r are at distance less or equal than y from f^μ_r.

Suppose that $x, y \neq 0$. Consider a path γ_l of minimum length from a face $f \in F_l$ to f^μ_l. Suppose that $f^\mu_r \in \gamma_l$. Then all faces in F_r have a path to f_l passing by f^μ_r of length less or equal than x. It follows that F_l and F_r may be replaced by $F_l \cup F_r$ and $\emptyset$, respectively, satisfying the pair $\langle x, 0\rangle$. This is a contradiction since $\hat{A}(\nu|\Gamma^1_\mu)$ is succinct and contains $p = \langle x, y\rangle$. Hence, no γ_l contains f^μ_r. Analogously, no path γ_r of minimum length from a face $f \in F_r$ to f^μ_r contains f^μ_l. It follows that $p \in A(\nu)$ and, since, by construction,



(a) (b) (c)

Figure 11: (a) Component R_1 satisfies pair $p^k = \langle x_k, y_k \rangle = \langle 1, 2 \rangle$. (b) Edge $(5, 7)$ has thickness 1. (c) Component R_1 satisfies, nested into $\Gamma^j_{S_2}$, pair $p^k_{first} = \langle 0, max(y_k, x_k + t(\mu)) \rangle = \langle 0, max(2, 1 + 1) \rangle = \langle 0, 2 \rangle$. Faces in F^k_l are filled with vertical lines and faces in F^k_r are filled with horizontal lines.

$S \preceq \hat{A}(\nu)$ and, by definition, $\hat{A}(\nu) \preceq A(\nu)$, there exists a pair $p' \in S$ such that $p' \preceq p$, contradicting the hypothesis.

Suppose $p \in \hat{A}(\nu|\Gamma^1_\mu)$ and $p = \langle x, 0 \rangle$. If $p \in \hat{A}(\nu)$ then, by construction, there exists a $p' \in S$ such that $p' \preceq p$, contradicting the hypothesis. If $p \notin \hat{A}(\nu)$, then some γ_l passes by f^μ_r and traverses a child of μ different from ν to end in f^μ_l. It follows that the pair $p' = \langle x, y \rangle$ with $y = x - t(\mu)$ belongs to $A(\nu)$. Hence, by construction, there exists a pair $\overline{p} \in S$ that precedes $\overline{p}_{last} \preceq p'' = \langle max(x, y + t(\mu)), 0 \rangle = \langle max(x, x), 0 \rangle = \langle x, 0 \rangle = p$, contradicting the hypothesis that there is no pair in S preceding p.

Analogous considerations show that if $p = \langle 0, y \rangle$, then there exists a pair $\overline{p} \in S$ that precedes $\overline{p}_{first} \preceq p$, contradicting the hypothesis. □

Lemma 10 *Starting from $\hat{A}(\nu_i)$ and $t(\mu)$, Algorithm* Nested_Series *computes $\hat{A}(\nu_i|\Gamma^1_\mu)$ in time $O(n(\nu_i))$.*

Proof: By Property 10, $S \preceq A(\nu_i|\Gamma^1_\mu)$. By Property 9, $S \subseteq A(\nu_i|\Gamma^1_\mu)$. By $S \preceq A(\nu_i|\Gamma^1_\mu)$ and $S \subseteq A(\nu_i|\Gamma^1_\mu)$, we have that S reduces $A(\nu_i|\Gamma^1_\mu)$. The statement follows from the fact that S reduces $A(\nu_i|\Gamma^1_\mu)$ and, by Property 8, S is succinct.

The time complexity bound follows from the fact that the construction of the pairs $\overline{p}_{first}$ and $\overline{p}_{last}$, as well as their insertion into S, is performed in linear time with respect to the cardinality of $\hat{A}(\nu_i)$ because in both cases each element of the set is considered only once. □

Lemma 11 *Starting from $\hat{A}(\nu_i)$ and $t(\nu_i)$, for $i = 1, \ldots, \delta(\mu)$, the gist $\hat{A}(\mu)$ can be computed in time $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$.*

Proof: By Lemma 9, thickness $t(\mu)$ can be computed in $O(\delta(\mu))$ time. By Lemma 10, given $\hat{A}(\nu_i)$ and $t(\mu)$, the gists $\hat{A}(\nu_i|\Gamma^1_\mu)$ of the admissible sets of ν_i nested into Γ^1_μ can be computed in $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$ total time. The intersection of such sets can still be computed in $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$ time, by Lemma 4. $\square$

4.1.2 The Rigid Case.

In the rigid case, since $sk(\mu)$ is a 3-connected component, it admits exactly two embeddings, Γ^1_μ and Γ^2_μ, which only differ for a flipping around its poles. In Fig. 12.a and Fig. 12.b embeddings $\Gamma^1_{R_1}$ and $\Gamma^2_{R_1}$ are shown, respectively. Hence, it is possible to consider one of the two embeddings only, say Γ^1_μ, compute the admissible set $A^1(\mu)$ of μ restricted to Γ^1_μ, and obtain the admissible set $A^2(\mu)$ of μ restricted to Γ^2_μ by swapping, for each pair of $A^1(\mu)$, elements x and y. The gist $\hat{A}(\mu)$ is given by the union of the two sets. By Lemma 8, $A^1(\mu)$ can be obtained by intersecting the gist $\hat{A}(\Gamma^1_\mu)$ of the admissible set of $sk(\mu)$ and the gists $\hat{A}(\nu_i|\Gamma^1_\mu)$ of the admissible sets of the components ν_i nested into Γ^1_μ. Fig. 12.c, where skeleton faces are drawn white and children faces ar drawn grey, shows an embedding of the pertinent graph G_{R_1} compatible with $\Gamma^1_{R_1}$.

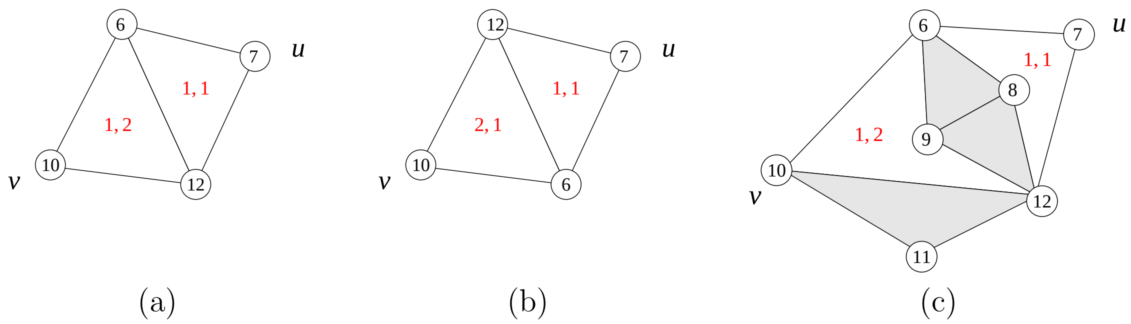


Figure 12: (a) Embedding $\Gamma^1_{R_1}$ of $sk(R_1)$. (b) Embedding $\Gamma^2_{R_1}$ of $sk(R_1)$. (c) An embedding of the pertinent graph G_{R_1} compatible with $\Gamma^1_{R_1}$. Skeleton faces are drawn white and children faces are drawn grey. Each skeleton face f is labeled with two integers representing its depths $d_l(f)$ and $d_r(f)$, respectively.

In order to compute $\hat{A}(\nu_i|\Gamma^1_\mu)$ and $\hat{A}(\Gamma^1_\mu)$ it is useful to label each face f of Γ^1_μ with its depths $d_l(f)$ and $d_r(f)$, as shown in Fig. 12. This can be done in linear time performing a single-source shortest path from the two external faces f^μ_l and f^μ_r.

We propose an algorithm, called NESTED_RIGID, which, given an R-node μ and one of its children ν, suitably builds a set S starting from $\hat{A}(\nu)$, $t(\mu)$, and the values of the depths $d_l(f)$ and $d_r(f)$ of each face f of Γ^1_μ. Then, we show that such a set is $\hat{A}(\nu|\Gamma^1_\mu)$. The algorithm first generates a set S' containing a pair $\langle x_k + d_l(f^\nu_l), y_k + d_r(f^\nu_r)\rangle$ for each pair $\langle x_k, y_k\rangle \in \hat{A}(\nu)$, and a set S'' containing a pair $\langle y_k + d_l(f^\nu_r), x_k + d_r(f^\nu_l)\rangle$ for each pair $\langle x_k, y_k\rangle \in \hat{A}(\nu)$. Then it initializes $S = S' \cup S''$. Observe that, by Property 6, S contains the two pairs $p_{first} = \langle 0, y_{max}\rangle$ and $p_{last} = \langle x_{max}, 0\rangle$.

For each pair $p^k = \langle x_k, y_k\rangle$ of $\hat{A}(\nu)$ define $p^k_{first} = \langle 0, max(y_k, x_k + t(\mu))\rangle$. Denote by $\overline{p}_{first}$ the p^k_{first} with minimum y and by $\overline{p}_{last}$ the pair obtained from $\overline{p}_{first}$ swapping the two elements x and y. If $\overline{p}_{first} \preceq p_{first}$, then insert $\overline{p}_{first}$ into S and remove from S any pair p^* such that $\overline{p}_{first} \preceq p^*$. If $\overline{p}_{last} \preceq p_{last}$, then append $\overline{p}_{last}$ to S and remove from S any pair p^* such that $\overline{p}_{last} \preceq p^*$.

Algorithm 4 Nested_Rigid

Input: An R-node μ, with its thickness $t(\mu)$ and the values of the depths $d_l(f)$ and $d_r(f)$ of each face of the embedding Γ_μ^1 of $sk(\mu)$, and one of its children ν, with its gist $\hat{A}(\nu)$.

Output: The gist $\hat{A}(\nu|\Gamma_\mu^1)$ of the admissible set of ν nested into Γ_μ^1.

for all $p^k = \langle x_k, y_k \rangle \in \hat{A}(\nu)$ **do**
    S'.addLast($\langle x_k + d_l(f_l^\nu), y_k + d_r(f_r^\nu)\rangle$);
    S''.addLast($\langle y_k + d_l(f_r^\nu), x_k + d_r(f_l^\nu)\rangle$);
    $p^k_{first} = \langle 0, \max\{x_k + d_r(f_l^\nu), y_k + d_r(f_r^\nu)\}\rangle$;
    $p^k_{last} = \langle \max\{x_k + d_l(f_l^\nu), y_k + d_l(f_r^\nu)\}, 0\rangle$;
    if $p^k_{first} \preceq \overline{p}_{first}$ **then**
        $\overline{p}_{first} = p^k_{first}$;
    end if
end for
$S = S' \cup S''$;
$p_{first} = S$.getFirst();
if $\overline{p}_{first} \preceq p_{first}$ **then**
    S.addFirst($\overline{p}_{first}$);
    $\overline{p}_{last} = \overline{p}_{first}$.swapElements();
    S.addLast($\overline{p}_{last}$);
    for all $p^* \neq \overline{p}_{first} \in S$ and $p^* \neq \overline{p}_{last} \in S$ **do**
        if $\overline{p}_{first} \preceq p^*$ or $\overline{p}_{last} \preceq p^*$ **then**
            S.remove(p^*);
        end if
    end for
end if
return S;

Let μ be an R-node with children ν_i, for $i = 1, \ldots, \delta(\mu)$ and let $n(\nu_i)$ be the number of vertices of ν_i.

Lemma 12 *The thickness $t(\mu)$ can be computed in $O(\delta(\mu))$ time.*

Proof: As shown in Fig. 4, the (u, v)-dual graph of $sk(\mu)$ is a triconnected component with the two vertices corresponding to f_l^μ and f_r^μ on the external face. Since, by definition, $t(\mu)$ is the distance between f_l^μ and f_r^μ in such a (u, v)-dual graph, which can be computed in $O(\delta(\mu))$ performing a shortest path algorithm between f_l^μ and f_r^μ, the statement follows. □

Property 11 *The set S computed by Algorithm* NESTED_RIGID *is succinct.*

Proof: S' and S'' are succinct since they are built adding the same constant values $d_l(f_l^\nu), d_l(f_r^\nu), d_r(f_l^\nu)$ and $d_r(f_r^\nu)$ to all the pairs of $\hat{A}(\nu)$, which is succinct and assumed ordered wrt the $\preceq_x$ relationship. $S' \cup S''$, computed with Algorithm GIST_UNION, is succinct by Lemma 7. Hence, S is succinct since it is initialized with a succinct set and, when the two pairs $\overline{p}_{first}$ and $\overline{p}_{last}$ are eventually added, the pairs p^* preceded by them are removed. □

Property 12 *The set S computed by Algorithm* NESTED_RIGID *is a subset of $A(\nu|\Gamma_\mu^1)$.*

Proof: Consider a pair $p^k = \langle x_k, y_k \rangle \in \hat{A}(\nu)$. There exists an embedding Γ_ν^k such that the set of the internal faces of Γ_ν^k can be partitioned into two sets F_l^k and F_r^k such that all faces in F_l^k have distance from f_l^ν less or equal than x_k and all faces in F_r^k have distance from f_r^ν less or equal than y_k. Consider any embedding $\Gamma_{G_\mu}^*$ such that $\Gamma_{G_\mu}^*$ restricted to ν is Γ_ν^k.

Consider a pair $p \in S$. Four are the cases: $p \in S'$, $p \in S''$, $p = \overline{p}_{first}$ and $p = \overline{p}_{last}$. If $p \in S'$ then $p = \langle x_k + d_l(f_l^\nu), y_k + d_r(f_r^\nu) \rangle$ for some k. We show that $p \in \hat{A}(\nu|\Gamma_\mu^1)$ as follows. Consider the faces of the embedding $\Gamma_{G_\mu}^*$ internal to ν, that is, the faces corresponding to Γ_ν^k, and their partition into F_r^k and F_l^k satisfying pair $\langle x_k, y_k \rangle$. For example, consider the partition of the internal faces of component R_2, shown in Fig. 13.a, satisfying pair $\langle 0, 1 \rangle$. As shown in Fig. 13.b, faces $f \in F_l^k$ are at distance less or equal than $x_k + d_l(f_l^\nu)$ from f_l^μ, since they are at distance less or equal than x_k from f_l^ν which, in turn, is at distance $d_l(f_l^\nu)$ from f_l^μ. Faces $f \in F_r^k$ are at distance less or equal than $y_k + d_r(f_r^\nu)$ from f_r^μ, since they are at distance less or equal than y_k from f_r^ν which, in turn, is at distance $d_r(f_r^\nu)$ from f_r^μ. Analogous considerations, also shown in Fig. 13.c, prove that if $p \in S''$ then $p \in A(\nu|\Gamma_\mu^1)$.

If $p = \overline{p}_{first}$ then $p = \langle 0, max(x_k + d_r(f_l^\nu), y_k + d_r(f_r^\nu)) \rangle$ for some k. We show that $p \in \hat{A}(\nu|\Gamma_\mu^1)$ as follows. Consider the faces of the embedding $\Gamma_{G_\mu}^*$ internal to ν Faces $f \in F_l^k$ are at distance less or equal than $x_k + d_r(f_l^\nu)$ from f_r^μ since they are at distance less or equal than x_k from f_l^ν which, in turn, is at distance $d_r(f_l^\nu)$ from f_r^μ. Faces $f \in F_r^k$ are at distance less or equal than $y_k + d_r(f_r^\nu)$ from f_r^μ since they are at distance less or equal than y_k from f_r^ν which, in turn, is at distance $d_r(f_r^\nu)$ from f_r^μ. Analogous considerations prove that if $p = \overline{p}_{last}$ then $p \in A(\nu|\Gamma_\mu^1)$. □

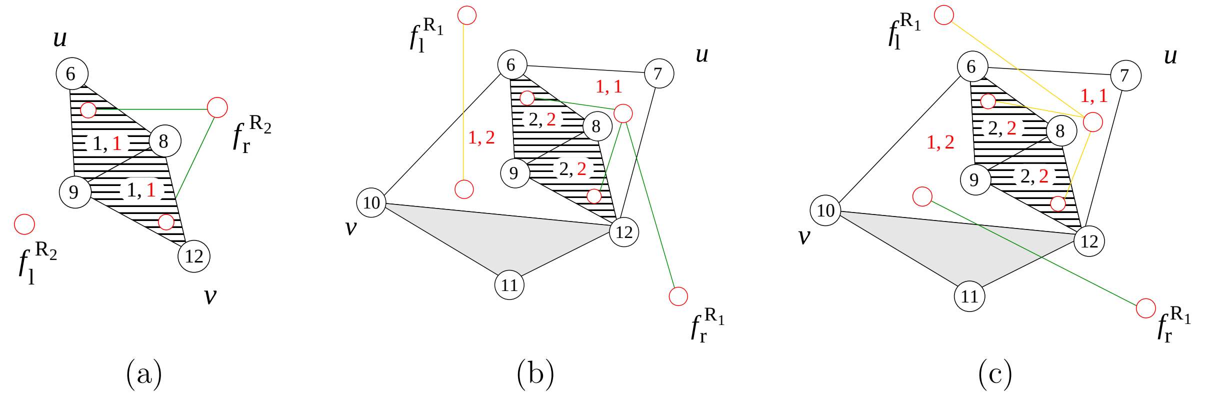


Figure 13: (a) Component R_2 satisfies pair $p^k = \langle x_k, y_k \rangle = \langle 0, 1\rangle$. (b) Component R_2 satisfies, nested into $\Gamma^1(R_1)$, pair $\langle x_k + d_l(f_l^\nu), y_k + d_r(f_r^\nu)\rangle = \langle 0+1, 1+1\rangle = \langle 1, 2\rangle$. (c) Component R_2 satisfies, nested into $\Gamma^1(R_1)$, pair $\langle y_k + d_l(f_r^\nu), x_k + d_r(f_l^\nu)\rangle = \langle 1+1, 0+2\rangle = \langle 2, 2\rangle$. Skeleton faces are drawn white, children faces not belonging to R_2 are drawn grey, faces in F_r^k are filled with horizontal lines.

Property 13 *The set S computed by Algorithm* NESTED_RIGID *precedes $A(\nu|\Gamma^1_\mu)$.*

Proof: Since $\hat{A}(\nu|\Gamma^1_\mu) \preceq A(\nu|\Gamma^1_\mu)$, it is sufficient to show that $S \preceq \hat{A}(\nu|\Gamma^1_\mu)$. Suppose for contradiction that there exists a pair $p = \langle x, y\rangle$ such that $p \in \hat{A}(\nu|\Gamma^1_\mu)$ and there is not a pair $p' \in S$ such that $p' \preceq p$. Since $p \in \hat{A}(\nu|\Gamma^1_\mu)$, by definition there exists an embedding Γ_{G_μ}, compatible with Γ^1_μ, such that all faces of ν can be partitioned into two sets F_l and F_r such that all faces in F_l are at distance from f_l^μ less or equal than x and all faces in F_r are at distance from f_r^μ less or equal than y.

Suppose that $x, y \neq 0$. Then $F_l, F_r \neq \emptyset$. Consider paths of minimum length $\gamma_l(f)$ from each face $f \in F_l$ to f^l_μ, and $\gamma_r(f')$ from each face $f' \in F_r$ to f^r_μ. By minimality of the paths, if $f^l_\nu \in \gamma_l(f)$ for each f, then $f^r_\nu \in \gamma_r(f')$ for each f', and if $f^r_\nu \in \gamma_l(f)$ for each f, then $f^l_\nu \in \gamma_r(f')$ for each f'. It follows that either $p' = \langle x - d_l(f_l^\nu), y - d_r(f_r^\nu)\rangle \in A(\nu)$ or $p' = \langle y - d_r(f_l^\nu), x - d_l(f_r^\nu)\rangle \in A(\nu)$. Consider $p' = \langle x - d_l(f_l^\nu), y - d_r(f_r^\nu)\rangle$; hence, by construction, there exists a pair $\overline{p} \in S$ that precedes a pair $p^k \in S'$ such that $p^k = \langle x_k + d_l(f_l^\nu), y_k + d_r(f_r^\nu)\rangle = \langle x - d_l(f_l^\nu) + d_l(f_l^\nu), y - d_r(f_r^\nu) + d_r(f_r^\nu)\rangle = \langle x, y\rangle$, contradicting the hypothesis that there is no pair in S preceding p. If $p' = \langle y - d_r(f_l^\nu), x - d_l(f_r^\nu)\rangle$, analogous considerations show that there exists a pair $\overline{p} \in S$ that precedes a pair $p^k \in S''$ such that $p^k = p$, contradicting the hypothesis.

Suppose $p = \langle x, 0\rangle$. Consider the partition of internal faces of ν into two sets F'_l and F'_r such that $f \in F'_l$ if $f^l_\nu \in \gamma_l(f)$ and $f^r_\nu \notin \gamma_l(f^l_\nu)$, and $f \in F'_r$ in the other cases. In such a partition faces in F'_l are at distance less or equal than $x - d_l(f_l^\nu)$ from f_l^ν and faces in F'_r are at distance less or equal than $x - d_l(f_r^\nu)$ from f_r^μ. It follows that pair $p' = \langle x - d_l(f_l^\nu), x - d_l(f_r^\nu)\rangle$ belongs to $A(\nu)$. Hence, by construction, there exists a pair $\overline{p} \in S$ that precedes $\overline{p}_{last} \preceq p'' = \langle max(x_k + d_l(f_l^\nu), y_k + d_l(f_r^\nu)), 0\rangle = \langle max(x - d_l(f_l^\nu) + d_l(f_l^\nu), x - d_l(f_r^\nu) + d_l(f_r^\nu)), 0\rangle = \langle \max(x, x), 0\rangle = \langle x, 0\rangle$, contradicting the hypothesis that there is no pair in S preceding p.

Analogous considerations show that, if $p = \langle 0, y\rangle$, then there exists a pair $\overline{p} \in S$ that precedes $\overline{p}_{first} \preceq p$, contradicting the hypothesis. $\square$

Lemma 13 *Starting from* $\hat{A}(\nu_i)$ *and* $t(\mu)$*, Algorithm* NESTED_RIGID *computes* $\hat{A}(\nu_i|\Gamma^1_\mu)$ *in time* $O(n(\nu_i))$*.*

Proof: By Property 13, $S \preceq A(\nu|\Gamma^1_\mu)$. By Property 12, $S \subseteq A(\nu|\Gamma^1_\mu)$, hence $A(\nu|\Gamma^1_\mu) \preceq S$. By $S \preceq A(\nu|\Gamma^1_\mu)$, $A(\nu|\Gamma^1_\mu) \preceq S$ and $S \subseteq A(\nu|\Gamma^1_\mu)$, S reduces $A(\nu|\Gamma^1_\mu)$. The statement follows from the fact that S reduces $A(\nu|\Gamma^1_\mu)$ and, by Property 11, S is succinct.

Now we show that Algorithm NESTED_RIGID runs in $O(n(\nu))$ time. The construction of the sets S' and S'' can be performed in $O(n(\nu))$ time because, for each pair of $\hat{A}(\nu)$, we have to perform $O(1)$ operations only. The union between S' and S'', computed with Algorithm GIST_UNION, is performed, by Lemma 7, in $O(n(\nu))$ time since $|S'| = |S''| = O(n(\nu))$. The construction of the pairs $\overline{p}_{first}$ and $\overline{p}_{last}$, as well as their insertion into S, is performed in $O(n(\nu))$ time because in both cases each element of $\hat{A}(\nu)$ is considered only once. $\square$

We present an algorithm, called SKELETON_RIGID, which, given an R-node μ with the values of the depths $d_l(f)$ and $d_r(f)$ of each face $f \in \Gamma^1_\mu$, suitably builds a set S that is the gist $\hat{A}(\Gamma^1_\mu)$ of the admissible set of $sk(\mu)$. The algorithm initializes $S = \emptyset$. Then, for increasing values of x, consider the partition of the internal faces of Γ^1_μ into the two sets F_l and F_r such that all faces at distance less or equal than x from f^μ_l are into F_l and the other ones are into F_r. Hence, a pair $\langle x, y\rangle$ such that all faces in F_r are at distance less or equal than y from f^μ_r is created and inserted into S if it is incomparable with the last pair of S.

Algorithm 5 SKELETON_RIGID

Input: An R-node μ, with the values of the depths $d_l(f)$ and $d_r(f)$ of each face of the embedding Γ^1_μ of $sk(\mu)$.

Output: The gist $\hat{A}(\Gamma^1_\mu)$ of the admissible set of $sk(\mu)$.

```
for x = 0 to max{d_l(f) | f ∈ F} do
    p = ⟨x, max{d_r(f) | f ∈ F & d_l(f) > x}⟩;
    if S.getLast() ≁ p then
        S.addLast(p);
    end if
end for
return S;
```

Lemma 14 *Starting from the values of the depths* $d_l(f)$ *and* $d_r(f)$ *of each skeleton face* f*, Algorithm* SKELETON_RIGID *computes the gist* $\hat{A}(\Gamma^1_\mu)$ *of the admissible set of* $sk(\mu)$ *in* $O(\delta(\mu))$ *time.*

Proof: To prove the correctness of the algorithm, we have to show that $S \subset A(\Gamma^1_\mu)$ and $S \preceq A(\Gamma^1_\mu)$. The first part is by construction, since each pair $p \in S$ is built considering a particular partition of the internal faces of Γ^1_μ into the two sets F_l and F_r. To show the second part, suppose, for contradiction, that there exists a pair $p' = \langle x', y'\rangle \in A(\Gamma^1_\mu)$ and there exists no pair $p \in S$ such that $p \preceq p'$. If $x' > \max\{d_l(f)|f \in F\}$, consider the pair $p = \langle x, y\rangle$ such that $x = \max\{d_l(f)|f \in F\}$. Since, by construction, all the internal faces are in F_l and F_r is $\emptyset$, we have $y = 0$. Hence, $p \preceq p'$ since $x < x'$ and $y \leq y'$. Since, by construction, we have that either $p \in S$ or there exists a pair $p'' \in S$ such that $p'' \preceq p \preceq p'$, we have a contradiction. If $x' \leq \max\{d_l(f)|f \in F\}$, consider the pair $p = \langle x, y\rangle$ with $x = x'$ and consider the two partitions F_l, F_r and F_l', F_r' satisfying p and p', respectively. We have $F_r \subseteq F_r'$, since, by construction, all the faces in F_r are at distance greater than x from f_l^{μ}. Hence, $y \leq y'$ and, since $x = x'$, we have $p \preceq p'$. Since, by construction, we have that either $p \in S$ or there exists a pair $p'' \in S$ such that $p'' \preceq p \preceq p'$, we have a contradiction again. The time complexity bound follows from the fact that Algorithm SKELETON_RIGID iterates $\max\{d_l(f)|f \in F\}$ times, which, by Property 7, is $O(\delta(\mu))$, and each iteration is executed in $O(1)$ time. □

Lemma 15 *Starting from $\hat{A}(\nu_i)$ and $t(\nu_i)$, for $i = 1, \ldots, \delta(\mu)$, the gist $\hat{A}(\mu)$ can be computed in time $O(\sum_{j=1}^{\delta(\mu)} \sum_{i=1}^{j} n(\nu_i))$.*

Proof: By Lemma 12, thickness $t(\mu)$ can be computed in $O(\delta(\mu))$ time. Given the embedding Γ_μ^1 of $sk(\mu)$ the gist $\hat{A}(\Gamma_\mu^1)$ can be computed, by Lemma 14, in $O(\delta(\mu))$ time. By Lemma 13, the gists $\hat{A}(\nu_i|\Gamma_\mu^1)$ of the admissible sets of ν_i nested into Γ_μ^1 can be computed in $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$ total time. The admissible set $A^1(\mu) = \hat{A}(\nu_i|\Gamma_\mu^1) \cap \hat{A}(\Gamma_\mu^1)$ restricted to the embedding Γ_μ^1 can be computed in $O(\sum_{j=1}^{\delta(\mu)} \sum_{i=1}^{j} n(\nu_i))$ time, according to Lemma 4. The admissible set $A^2(\mu)$ restricted to the other embedding can be computed in $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$ time creating, for each pair $\langle x, y\rangle \in A^j(\mu)$, a pair $\langle y, x\rangle$. The gist $\hat{A}(\mu) = A^1(\mu) \cup A^2(\mu)$ is computed in $O(\sum_{i=1}^{\delta(\mu)} n(\nu_i))$ time, according to Lemma 7. □

4.1.3 The Parallel Case.

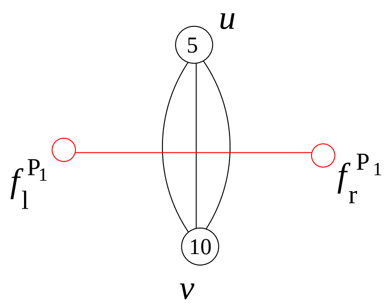


Figure 14: One of the embeddings of $sk(P_1)$ and its (u, v)-dual.

In the parallel case, as shown in Fig. 14, $sk(\mu)$ is composed of two vertices, u and v, with $\delta(\mu)$ parallel edges between them, and admits a factorial number of embeddings, that is, the number of all possible permutations of its $\delta(\mu)$ edges. Hence, according to Lemma 8, the gist $\hat{A}(\mu)$ can be obtained by performing the union between $\delta(\mu)!$ sets, where each set $A^j(\mu)$ corresponds to a different embedding Γ_μ^j of $sk(\mu)$. Also, $A^j(\mu)$ can be computed by intersecting $\hat{A}(\Gamma_\mu^j)$ and the gists $\hat{A}(\nu_i|\Gamma_\mu^j)$ of the admissible sets of the components ν_i nested into Γ_μ^j. Hence, a naïve computation of $\hat{A}(\mu)$ employs a factorial number of steps. We reduce the number of permutations to be analyzed by exploiting the following properties and considerations.

Let μ be a P-node with children ν_i, for $i = 1, \ldots, \delta(\mu)$. First, consider a pair $\langle x, y\rangle \in \hat{A}(\mu)$ and an embedding Γ_{G_μ} of G_μ satisfying $\langle x, y\rangle$, i.e., whose internal faces can be partitioned into two sets F_l and F_r such that all faces in F_l (F_r) have distance from f_l^μ (f_r^μ) less or equal than x (y).

Lemma 16 *The thickness $t(\mu) = \sum_{i=1}^{\delta(\mu)} t(\nu_i)$ can be computed in $O(\delta(\mu))$ time.*

Proof: As shown in Fig. 14, the (u, v)-dual graph of each embedding of $sk(\mu)$ is made up of a single path of length $\delta(\mu)$ connecting the two vertices corresponding to f_l^μ and f_r^μ and passing for all the internal vertices. Since, by definition, $t(\mu)$ is the distance between f_l^μ and f_r^μ in such a (u, v)-dual graph, the statement follows. □

Let μ be a P-node and let Γ_{G_μ} be an embedding of G_μ satisfying pair $\langle x, y\rangle \in \hat{A}(\mu)$. Consider the partition of the set of the internal faces of Γ_{G_μ} into the two sets F_l and F_r such that all faces in F_l have distance from f_l^μ less or equal than x and all faces in F_r have distance from f_r^μ less or equal than y. Given a face $f \in \Gamma_{G_\mu}$, we denote by $\gamma_l(f)$ the path of minimum length from f to f_l^μ, and by $\gamma_r(f)$ the path of minimum length from f to f_r^μ. The following properties hold.

Property 14 *Let ν be one of the children of μ. If the children faces of Γ_{G_μ} corresponding to the internal faces of ν are split by F_l and F_r, then, for each face $f \in F_l$, $f_l^\nu \in \gamma_l(f)$ and $f_r^\nu \notin \gamma_l(f_l^\nu)$. Analogously, for each face $f \in F_r$, $f_r^\nu \in \gamma_r(f)$ and $f_l^\nu \notin \gamma_r(f_r^\nu)$.*

Proof: Consider the sequence of skeleton faces $f_l^\mu = f_1, f_2, \ldots, f_{\delta(\mu)}, f_{\delta(\mu)+1} = f_r^\mu$. Distances $d_l(f_i)$, with $i = 1, \ldots, \delta(\mu) + 1$ are increasing, while $d_r(f_i)$ are decreasing. Hence, $d_l(f_l^\nu) < d_l(f_r^\nu)$ and $d_r(f_l^\nu) > d_r(f_r^\nu)$. Consider a face $f \in F_l$. Since $\gamma_l(f)$ is a path of minimum length and $d_l(f_l^\nu) < d_l(f_r^\nu)$, the statement follows. The same for a face $f \in F_r$.
□

Property 15 *Let f_{sk} be a skeleton face of Γ_{G_μ} and let ν_l and ν_r be two child components of μ, incident to f_{sk}, such that $f_r^{\nu_l} = f_{sk} = f_l^{\nu_r}$. If some internal face f_l of ν_l belongs to F_r and some internal face f_r of ν_r belongs to F_l, then there exists a partition F_l' and F_r' such that all faces in F_l' have distance from f_l^μ less or equal than x and all faces in F_r' have distance from f_r^μ less or equal than y and such that either all internal faces of ν_l belong to F_l or all internal faces of ν_r belong to F_r.*

Proof: By Property 14, we have $f_{sk} = f_r^{\nu_l} \in \gamma_r(f_l)$, for each face f_l of ν_l, and $f_{sk} = f_l^{\nu_r} \in \gamma_l(f_r)$, for each face f_r of ν_r. Denote by $d_l'(f)$ and $d_r'(f)$ the distance from an internal face f of ν_i, with $i \in \{l, r\}$, to $f_l^{\nu_i}$ and $f_r^{\nu_i}$, respectively. Denote by $d_l'(\nu_i)$ the maximum $d_l'(f)$ such that $f \in F_l$ and f is internal to ν_i. The same for $d_r'(\nu_i)$. If $d_l'(\nu_r) < d_r'(\nu_l)$, then consider the partition F_l' and F_r' obtained from F_l and F_r moving the internal faces f_r of ν_r from F_l to F_r. Then, all faces in F_l' have distance from f_l^μ less or equal than x, since $F_l' \subset F_l$, and all faces in F_r' have distance from f_r^μ less or equal than y, since, by hypothesis, faces $f \in F_r$ are at distance less or equal than y from f_r^μ and each face f_r internal to ν_r is at distance $d_l'(f_r) + d_r(f_{sk}) < d_l'(f_r) + d_r(f_{sk}) \leq y$ from f_r^μ. Analogously, if $d_l'(\nu_r) > d_r'(\nu_l)$, then the partition F_l' and F_r' obtained from F_l and F_r moving the internal faces f_l of ν_l from F_r to F_l is such that all faces in F_l' have distance from f_l^μ less or equal than x and all faces in F_r' have distance from f_r^μ less or equal than y. Fig. 15 shows the case $d_l'(\nu_r) > d_r'(\nu_l)$. □

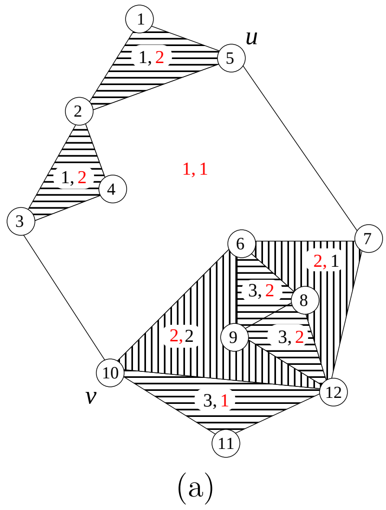


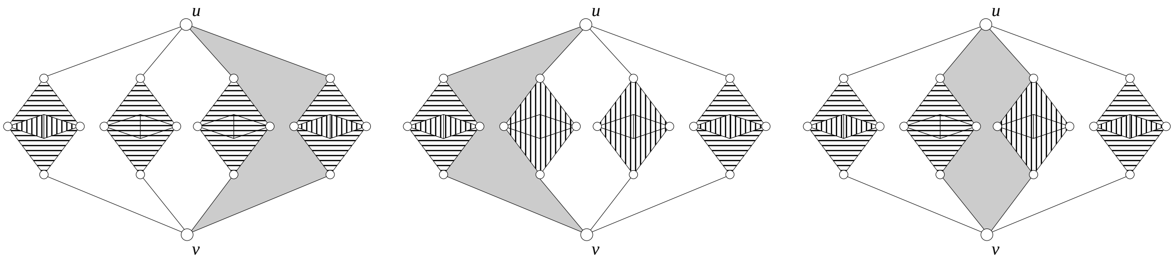
Figure 15: Property 15. Component S_1 is ν_l and component S_2 is ν_r. (a) Parallel component $\{S_1, S_2\}$ satisfies pair $\langle 2, 2\rangle$ when some faces of S_1 are in F_r and some faces of S_2 are in F_l. (b) Parallel component $\{S_1, S_2\}$ satisfies pair $\langle 0, 2\rangle \preceq \langle 2, 2\rangle$ when all faces of S_2 are in F'_r. Faces in F_l are filled with vertical lines, faces in F_r and F'_r are filled with horizontal lines, and face f_{sk} is drawn white.

Lemma 17 *Let Γ_{G_μ} be an embedding of G_μ satisfying pair $\langle x, y\rangle \in \hat{A}(\mu)$. There exists a partition F'_l and F'_r such that:*

1. *All faces in F'_l have distance from f^μ_l less or equal than x*
2. *All faces in F'_r have distance from f^μ_r less or equal than y*
3. *There exists at most one component ν_c whose internal faces belong to both the sets F'_l and F'_r*
4. *Each child component at the left of ν_c has its internal faces in F'_l and each child component at the right of ν_c has its internal faces in F'_r*

Proof: If there exist in Γ_{G_μ} two child components ν_1 and ν_2, possibly not sharing an external face, with ν_1 to the left of ν_2, such that some of the internal faces of ν_1 are in F_r and some of the internal faces of ν_2 are in F_l, then there is at least one skeleton face f_{sk} whose two incidents components ν_l and ν_r, with $f^{\nu_l}_r = f_{sk} = f^{\nu_r}_l$, have some internal face f_l of ν_l in F_r and some internal face f_r of ν_r in F_l, respectively. A proof of this fact is shown in Fig. 16, where the first component, ν_1, has some faces in F_r and the last component, ν_2, has some faces in F_l. Observe that we analyze only the case where components between ν_1 and ν_2 have all their internal faces in the same set since, in the other case, it is possible to find at least one subsequence of components where the above property is satisfied.

Hence, by Property 15, it is possible to find a partition F'_l and F'_r such that all faces in F'_l have distance from f^μ_l less or equal than x, all faces in F'_r have distance from f^μ_r less or equal than y and either all internal faces of ν_l belong to F_l or all internal faces of ν_r belong to F_r. The partition F'_l and F'_r can be modified as described above until Property 15 can not be further applied. Two are the cases: either there is a single child component ν_c



(a) (b) (c)

Figure 16: Proof of Lemma 17. Leftmost component is ν_1 and rightmost component is ν_2. Faces in F_l are filled with vertical lines, faces in F_r are filled with horizontal lines, f_{sk} is drawn dark grey, and the other skeleton faces are drawn white.

whose internal faces belong to both the sets F'_l and F'_r or there is none. In the first case, since Property 15 can not be applied, all child components to the left (right) of ν_c have their internal faces in F'_l (F'_r, respectively). In the second case, for analogous reasons, all child components with their faces in F'_l are to the left of those child components whose faces are in F'_r. □

The unique component ν_c, if any, whose faces belong to both F'_l and F'_r is called the *center* of the permutation. Intuitively, Lemma 17 states that we could restrict to consider those partitions F_l and F_r of the internal faces of Γ_{G_μ} such that each child component different from ν_c has its internal faces into the same set F_l or F_r. Consider, for example, embedding $\Gamma^j_{P_1}$ of parallel component P_1, shown in Fig. 17, corresponding to permutation $S_1, Q_{5,10}, S_2$, and consider $Q_{5,10}$ as the center of the permutation. It is possible to observe that a partition F_l, F_r where children faces of S_2, which is to the right of the center, are split by the two sets(Fig. 17.a) corresponds to a pair that is preceded by a pair corresponding to a partition where all faces of S_2 are in F_r(Fig. 17.b). In other words, for each child component ν_i different from ν_c, $\hat{A}(\nu_i)$ can be assumed to contain the two pairs $\langle x, 0\rangle$ and $\langle 0, y\rangle$ only.

The gist $\hat{A}(\mu)$ can be computed by choosing, one by one, each child component ν_c as the center of the permutation and inserting the other components either to the left or the right of ν_c, until a complete permutation is obtained. Each subsequence σ of components is associated with the gist $\hat{A}(\sigma)$ of its admissible set $A(\sigma)$, which is properly updated when a component is inserted. This approach would obtain the same permutation $\delta(\mu)$ times, exploring $O(\delta(\mu) \cdot \delta(\mu)!)$ sequences. Hence, at first glance, the computational complexity is augmented. However, we show in the following that focusing on ν_c can greatly help to reduce the number of permutations to be considered.

Lemma 18 *Let σ be a sequence of child components, with $\nu_c \in \sigma$, and let $\nu_i \notin \sigma$ be a child component of μ with $\langle 0, y_i\rangle, \langle x_i, 0\rangle \in \hat{A}(\nu_i)$. Adding ν_i to the left of σ we obtain a sequence σ'. Consider the set S' containing a pair $\langle \max(x + t(\nu_i), x_i), y\rangle$ for each pair $\langle x, y\rangle \in \hat{A}(\sigma)$. We have that $S' \preceq A(\sigma')$ and $S' \subset A(\sigma')$. Analogously, adding ν_i to the right of σ we obtain a sequence σ'' and the set S'' containing a pair $\langle x, \max(y + t(\nu_i), y_i)\rangle$ for each pair $\langle x, y\rangle \in \hat{A}(\sigma)$ is such that $S'' \preceq A(\sigma'')$ and $S'' \subset A(\sigma'')$.*

Proof: Consider a pair $\langle x, y\rangle \in \hat{A}(\sigma)$ and the embedding Γ_σ of σ such that there exists a partition of its internal faces into two sets F_l and F_r where faces in F_l are at distance

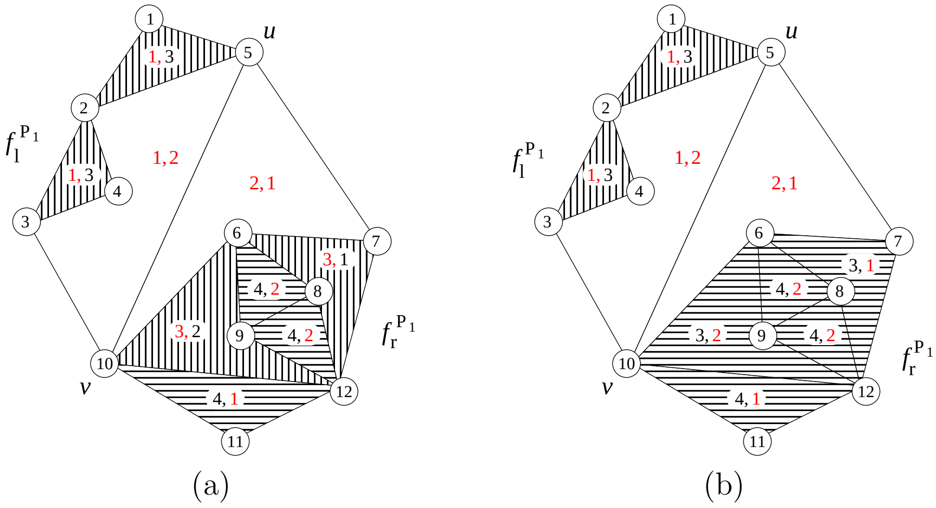


Figure 17: Embedding $\Gamma^j_{P_1}$ corresponds to permutation $S_1, Q_{5,10}, S_2$ and $Q_{5,10}$ is the center of the permutation. (a) $\Gamma^j_{P_1}$ satisfies pair $\langle 3, 2\rangle$ when children faces of S_2 are split by F_l and F_r. (b) $\Gamma^j_{P_1}$ satisfies pair $\langle 1, 2\rangle \preceq \langle 3, 2\rangle$ when all children faces of S_2 are in F_r. Skeleton faces are drawn white, faces in F_l are filled with vertical lines, and faces in F_r are filled with horizontal lines.

from f_l^σ less or equal than x, and faces in F_r are at distance from f_r^σ less or equal than y. Then, consider the embedding Γ_{ν_i} of ν_i such that all internal faces are at distance from $f_l^{\nu_i}$ less or equal than x_i. When inserting ν_i to the left of σ, consider the embedding $\Gamma_{\sigma'}$ of the new sequence σ' such that $\Gamma_{\sigma'} = \Gamma_\sigma \cup \Gamma_{\nu_i}$. There exists a partition of the internal faces of $\Gamma_{\sigma'}$ into two sets $F'_l = F_l \cup \Gamma_{\nu_i}$ and $F'_r = F_r$. Faces in F_l are at distance $x + t(\nu_i)$ from $f_l^{\sigma'}$ since they are at distance less or equal than x from f_l^σ which, in turn, is at distance $t(\nu_i)$ from $f_l^{\sigma'}$. Faces in Γ_{ν_i} are at distance less or equal than x from $f_l^{\sigma'}$ by construction. Hence, faces in F'_l are at distance from $f_l^{\sigma'}$ less or equal than $\max(x + t(\nu_i), x_i)$. Since $F'_r = F_r$ we have $\langle \max(x+t(\nu_i), x_i), y\rangle \in A(\sigma')$. Repeating such a procedure for each pair $\langle x, y\rangle \in \hat{A}(\sigma)$ we have that $S' \subset A(\sigma')$.

Suppose for contradiction that there exists a pair $p' = \langle x', y'\rangle$ such that $p' \in A(\sigma')$ and there is not a pair $p \in S'$ such that $p \preceq p'$. Since $p' \in A(\sigma')$, by definition, there exists an embedding $\Gamma_{\sigma'}$ whose internal faces can be partitioned into two sets F'_l and F'_r such that all faces in F'_l are at distance from $f_l^{\sigma'}$ less or equal than x' and all faces in F'_r are at distance from $f_r^{\sigma'}$ less or equal than y'. Consider the embeddings Γ_σ of σ and Γ_{ν_i} of ν_i induced by $\Gamma_{\sigma'}$. By Property 15, since $\nu_c \in \sigma$ and ν_i is to the left of σ, we have that Γ_{ν_i} has all its internal faces into F'_l. Hence, $x_i \leq x'$. Consider the partition of the internal faces of Γ_σ into the sets $F_l = F'_l \backslash \Gamma_{\nu_i}$ and $F_r = F'_r$. Faces in F_l are at distance $x' - t(\nu_i)$ from f_l^σ, which is at distance $t(\nu_i)$ from $f_l^{\sigma'}$, and faces in F_r are at distance y' from f_r^σ. Hence, embedding Γ_σ of σ satisfies pair $\langle x, y\rangle$, with $x = x' - t(\nu_i)$ and $y = y'$. Hence, by construction, S' contains a pair $p = \langle \max(x + t(\nu_i), x_i), y\rangle = \langle \max(x' - t(\nu_i) + t(\nu_i), x_i), y'\rangle = \langle \max(x', x_i), y'\rangle = \langle x', y'\rangle$, which is a contradiction.

Analogous considerations show that, when inserting ν_i to the right of σ, $S'' \preceq A(\sigma'')$ and $S'' \subset A(\sigma'')$. □

Let ν_i be a component. We introduce function $l(\nu_i) = t(\nu_i) - x_i$, where x_i is such that

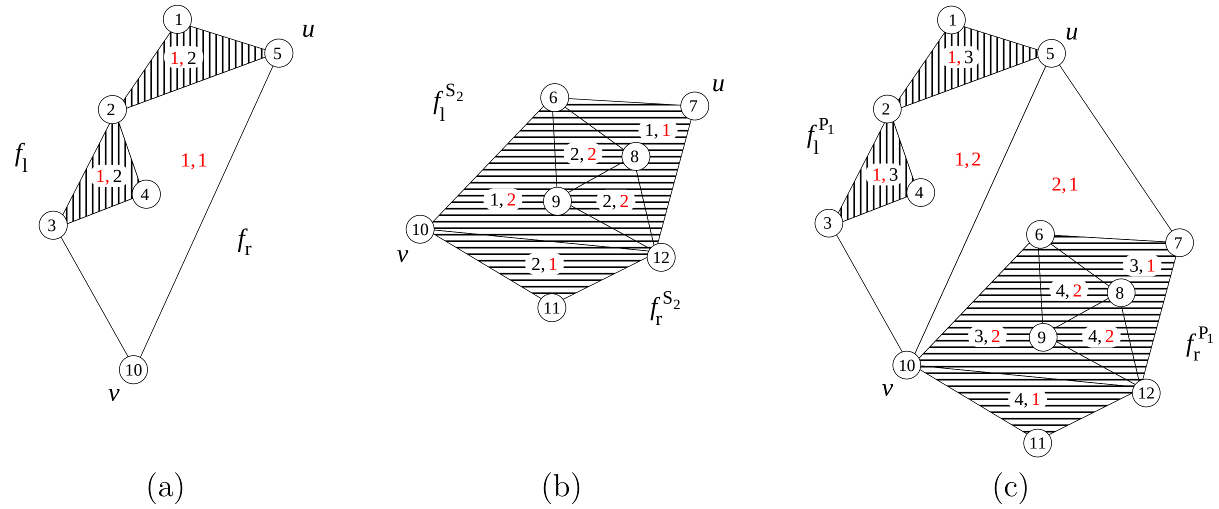


Figure 18: (a) Parallel component $\{S_1, Q_{5,10}\}$ satisfies pair $\langle 1, 0\rangle$. (b) Pair $\langle 0, 2\rangle \in \hat{A}(S_2)$. (c) After inserting S_2 to the right of the center, parallel component $P_1 = \{S_1, Q_{5,10}, S_2\}$ satisfies pair $\langle x_j, \max(y_j + t(\nu_i), y_i)\rangle = \langle 1, \max(0+1, 2)\rangle = \langle 1, 2\rangle$. Skeleton faces are drawn white, faces in F_l are filled with vertical lines, and faces in F_r are filled with horizontal lines.

pair $\langle x_i, 0\rangle \in \hat{A}(\nu_i)$.

Lemma 19 *Let σ be a sequence of child components of μ, with $\nu_c \in \sigma$, and let ν' and ν'' be two child components of μ, with $\nu', \nu'' \notin \sigma$. Let Γ^1_μ and Γ^2_μ be two embeddings of $sk(\mu)$ corresponding to two permutations of child components which only differ for the swapping of the two components ν' and ν'' lying on the same side of σ. If $l(\nu') < l(\nu'')$, then $A^2(\mu) \preceq A^1(\mu)$.*

Proof: Consider pairs $\langle x, y\rangle \in \hat{A}(\sigma)$, $\langle x', 0\rangle \in \hat{A}(\nu')$ and $\langle x'', 0\rangle \in \hat{A}(\nu'')$. First, consider embedding Γ^1_μ. When component ν' is inserted to the left of σ, by Lemma 18, we obtain a new sequence satisfying pair $\langle \max(x + t(\nu'), x'), y\rangle$. Then, when ν'' is inserted, we obtain a new sequence satisfying $p^1 = \langle \max(\max(x + t(\nu'), x') + t(\nu''), x''), y\rangle$. Second, consider embedding Γ^2_μ. In this case, after inserting ν'' and ν', we obtain a new sequence satisfying pair $p^2 = \langle \max(\max(x + t(\nu''), x'') + t(\nu'), x'), y\rangle$. Suppose, for contradiction, that $p^2 \npreceq p^1$. Suppose $x + t(\nu') > x'$. Then, since by hypothesis $t(\nu') - x' < t(\nu'') - x''$, we have $x + t(\nu'') > x''$. Hence, $p^1 = \langle \max(x + t(\nu') + t(\nu''), x''), y\rangle = \langle x + t(\nu') + t(\nu''), y\rangle$ and $p^2 = \langle \max(x + t(\nu'') + t(\nu'), x'), y\rangle = \langle x + t(\nu'') + t(\nu'), y\rangle = p^1$, contradicting the hypothesis that $p^2 \npreceq p^1$. Suppose $x + t(\nu') < x'$. Then, $p^1 = \langle \max(x' + t(\nu''), x''), y\rangle$. If $x + t(\nu'') > x''$ then $p^2 = \langle \max(x + t(\nu'') + t(\nu'), x'), y\rangle$. Hence, since $x' + t(\nu'') > x + t(\nu'') + t(\nu')$, by hypothesis, and $x' + t(\nu'') > x'$, by the non-negativity of the distance $t(\nu'')$, we have $p^2 \preceq p^1$, contradicting the hypothesis. If $x + t(\nu'') \leq x''$ then $p^2 = \langle \max(x'' + t(\nu'), x'), y\rangle$. Hence, since $x' + t(\nu'') > x'' + t(\nu')$, by $l(\nu') < l(\nu'')$, and $x' + t(\nu'') > x'$, we still have $p^2 \preceq p^1$, contradicting the hypothesis. Since for every pair $p^1 \in A^1(\mu)$ we have a pair $p^2 \in A^2(\mu)$ such that $p^2 \preceq p^1$, it follows that $A^2(\mu) \preceq A^1(\mu)$. □

Intuitively, any permutation with a component ν' further from ν_c than a second component ν'' with $l(\nu'') < l(\nu')$ can be ignored since its admissible set is preceded by that computed with another permutation. Therefore, the number of analyzed permutations can be reduced by ordering child components by decreasing values of $l(\nu)$ and, once the center ν_c of the permutation has been chosen, by adding the other components, ordered wrt function l, either to its left or to its right.

To do so, we build a rooted tree $T(\nu_c)$ of height $\delta(\mu)+1$ where each node p at distance d from the root is a pair $\langle x_p, y_p\rangle$ of non-negative integers and is associated with a sequence σ_p of d child components of μ such that $\langle x_p, y_p\rangle \in \hat{A}(\sigma_p)$. The nodes at distance d from the root are the incomparable pairs of integers satisfied by some sequence of length d. Hence, the set of nodes at distance $\delta(\mu)$ from the root is $\hat{A}(\mu)$ restricted to the permutations having ν_c as the center. Such a set is denoted by $\hat{A}_{\nu_c}(\mu)$.

Tree $T(\nu_c)$ is built as follows. The root is pair $\langle 0,0\rangle$ and is associated with the empty sequence. The root is added as many children as many pairs in the gist $\hat{A}(\nu_c)$ of the admissible set of the center of the permutation ν_c, each one associated with the sequence composed by ν_c only. The following levels are obtained by considering one by one the other components in decreasing order of l. When the k-th component ν_k is processed, each node p at depth $k-1$ is added two children p_l and p_r, corresponding to the sequences $\nu_k \cdot \sigma_p$ and $\sigma_p \cdot \nu_k$, respectively, obtained by adding ν_k to the left or to the right of σ_p. Pairs p_l and p_r are computed, starting from p, with the function presented in Lemma 18. From the set of pairs introduced at level k all those preceded by a pair of the same level can be removed, pruning the tree.

The gist $\hat{A}(\mu)$ of the admissible set of the P-node μ can be obtained as the union of the gists $\hat{A}_{\nu_c}(\mu)$, for each child component ν_i chosen as the center of the permutation ν_c.

Consider, for example, the P-node P_1 with child components $Q_{5,10}$, S_1, and S_2 shown in Fig. 18. We have $\langle 0,0\rangle \in \hat{A}(Q_{5,10})$ and $t(Q_{5,10}) = 1$, $\langle 1,0\rangle \in \hat{A}(S_1)$ and $t(S_1) = 1$, $\langle 0,2\rangle \in \hat{A}(S_2)$ and $t(S_2) = 1$. Hence, $l(Q_{5,10}) = 1$, $l(S_1) = 0$ and $l(S_2) = -1$. The ordering of the components is then $Q_{5,10} - S_1 - S_2$. Consider $Q_{5,10}$ as the center of the permutation. We have $\hat{A}(Q_{5,10}) = \{\langle 0,0\rangle\}$. The tree that computes the admissible set restricted to all the permutations having $Q_{5,10}$ as center is shown in Fig. 19.

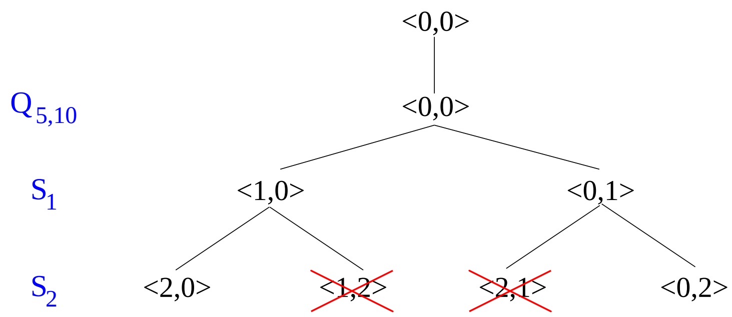


Figure 19: Computation of tree $T(Q_{5,10})$ for the P-node P_1 with child components $Q_{5,10}$, S_1, and S_2 inserted in this order. Pairs $\langle 1,2\rangle$ and $\langle 2,1\rangle$ have been removed since they are preceded by $\langle 0,2\rangle$ and $\langle 2,0\rangle$, respectively.

We propose an algorithm, called TREE_PARALLEL, which, given a P-node μ and its children ν_i, for $i = 1, \ldots, \delta(\mu)$, with one of them, ν_c, chosen as the center of the permutation, suitably builds a set S that is the gist $\hat{A}_{\nu_c}(\mu)$ of the admissible set restricted to all the permutations having ν_c as center. Components ν_i are considered ordered with respect to decreasing values of function $l(\nu_i)$. At every step of the computation, S contains the pairs associated to the nodes of the last level of the tree $T(\nu_c)$. S is initialized with $\hat{A}(\nu_c)$. When adding the k-th component ν_k, we first produce the two sets S_l and S_r of children obtained by concatenating ν_k to the left and to the right of σ_p, respectively, for each $p \in S$. Hence, pairs in S_l and S_r are created applying the function presented in Lemma 18 to each pair of S. Since S is ordered wrt the $\preceq_x$ relationship, and due to the formula used to compute nodes in S_l and S_r, these two sets can be kept ordered wrt the $\preceq_x$ relationship and succinct, by comparing the pair to be inserted only with the last pair. S is computed by performing the union between S_l and S_r as described in Lemma 7. The algorithm iterates until all the children components have been added.

Algorithm 6 TREE_PARALLEL

Input: A P-node μ, its children ν_i, for $i = 1, \ldots, \delta(\mu)$, with $t(\nu_i)$ and pairs $\langle x_i, 0\rangle, \langle 0, y_i\rangle \in \hat{A}(\nu_i)$, and one of the children, ν_c, chosen as the center of the permutation, with $\hat{A}(\nu_c)$

Output: The gist $\hat{A}_{\nu_c}(\mu)$ of the admissible set restricted to all the permutations having ν_c as center.

$S = \hat{A}(\nu_c)$;
for $i = 1$ to $\delta(\mu)$ **do**
  for all $p^k = \langle x_k, y_k\rangle \in S$ **do**
    if $p_l \not\preceq \langle \max\{x_k + t(\nu_i), x_i\}, y_k\rangle$ **then**
      $p_l = \langle \max\{x_k + t(\nu_i), x_i\}, y_k\rangle$;
      S_l.addLast(p_l);
    end if
    if $p_r \not\preceq \langle x_k, \max\{y_k + t(\nu_i), y_i\}\rangle$ **then**
      $p_r = \langle x_k, \max\{y_k + t(\nu_i), y_i\}\rangle$;
      S_r.addLast(p_r);
    end if
  end for
  $S = S_l \cup S_r$;
end for
return S;

Lemma 20 *Starting from $\hat{A}(\nu_i)$ and $t(\nu_i)$, Algorithm* TREE_PARALLEL *computes the gist $\hat{A}_{\nu_c}(\mu)$ of the admissible set of μ, restricted to all the permutations having ν_c as center, in $O(\sum_{k=1}^{\delta(\mu)} \sum_{i=1}^{k} n(\nu_i))$ time.*

Proof: When the k-th component is added, the computational complexity of the construction of S_l and S_r and of their union depends on the number of nodes at distance k from the root, which is, by definition, $O(\sum_{i=1}^{k} n(\nu_i))$. Since such operations have to be performed for each level, the overall time complexity bound follows. □

Lemma 21 *Starting from $\hat{A}(\nu_i)$ and $t(\nu_i)$, for $i = 1, \ldots, \delta(\mu)$, the gist $\hat{A}(\mu)$ of the admissible set of μ can be computed in time $O(\delta(\mu) \cdot \sum_{k=1}^{\delta(\mu)} \sum_{i=1}^{k} n(\nu_i))$.*

Proof: By Lemma 16, thickness $t(\mu)$ can be computed in $O(\delta(\mu))$ time. The gists $\hat{A}_{\nu_c}(\mu)$, for $c = 1, \ldots, \delta(\mu)$ of the admissible sets restricted to all the permutations having ν_c as center, can be computed, by Lemma 20, in $O(\delta(\mu) \cdot \sum_{k=1}^{\delta(\mu)} \sum_{i=1}^{k}(n(\nu_i)))$ total time. Since, for each gist $\hat{A}_{\nu_c}(\mu)$, we have $|\hat{A}_{\nu_c}(\mu)| = O(n)$, the gist $\hat{A}(\mu) = \cup_{c=1}^{\delta(\mu)} \hat{A}_{\nu_c}(\mu)$ of the admissible set of μ can be computed in $O(\delta(\mu) \cdot n)$ time, according to Lemma 7. □

4.2 Computing the Minimum Depth and the Minimum-Depth Embedding

At the end of the bottom-up traversal of $\mathcal{T}$, the value of the minimum depth can be computed starting from the gist of the admissible set of the component μ that is the child of the root e of $\mathcal{T}$. Namely, for each pair $\langle x_h, y_h \rangle \in \hat{A}(\mu)$, denote by m_h the maximum between x_h and y_h. The minimum depth is the minimum of the m_h.

Lemma 22 *Let μ be the child of the root of the SPQR-tree of an n-vertex graph G. Starting from $\hat{A}(\mu)$, the minimum depth of G can be computed in $O(n)$ time.*

Proof: Consider a pair $\langle x_h, y_h \rangle \in \hat{A}(\mu)$. By definition, there exists an embedding Γ_G^h such that, for each face $f \in \Gamma_G^h$, f is at distance less or equal than x_h from f_l^μ or f is at distance less or equal than y_h from f_r^μ. Since, by construction, f_l^μ and f_r^μ correspond to the same external face f^μ, we have that, for each face $f \in \Gamma_G^h$, f is at distance less or equal than $m_h = \max(x_h, y_h)$ from f^μ. Hence, m_h is the depth of the embedding satisfying $\langle x_h, y_h \rangle$. Since each pair $\langle x_h, y_h \rangle \in \hat{A}(\mu)$ corresponds to an embedding Γ_G^h, the minimum of the m_h is the minimum depth of G through all of its embeddings. The operations to be performed are $|\hat{A}(\mu)|$ maximums between two integers and a minimum among $|\hat{A}(\mu)|$ integers. Since, by Property 7, $|\hat{A}(\mu)|$ is $O(n)$, the statement follows. □

Theorem 1 *Let G be an n-vertex biconnected planar graph and let $\mathcal{T}$ be the SPQR-tree of G rooted at e. The minimum depth of an embedding of G with e on the external face can be computed in $O(n^3)$ time and $O(n^2)$ space.*

Proof: Consider the three sets S, R and P containing the series, rigid and parallel nodes of $\mathcal{T}$, respectively. For each series component $\mu_s \in S$, by Lemmas 9 and 11, $t(\mu_s)$ and $\hat{A}(\mu_s)$ can be computed in $O(\sum_{i=1}^{\delta(\mu_s)} n(\nu_i))$ time. Hence, the overall complexity for all the series nodes is $O(\sum_{\mu_s \in S} \sum_{i=1}^{\delta(\mu_s)} n(\nu_i))$. Since the number of the series nodes is $O(n)$, the above sum is $O(n^2)$.

For each rigid component $\mu_r \in R$, by Lemmas 12 and 15, $t(\mu_r)$ and $\hat{A}(\mu_r)$ can be computed in $O(\sum_{j=1}^{\delta(\mu_r)} \sum_{i=1}^{j} n(\nu_i))$. Hence, the overall complexity for all the rigid nodes is $O(\sum_{\mu_r \in R} \overbrace{\sum_{j=1}^{\delta(\mu_r)}}^{O(n)} \overbrace{\sum_{i=1}^{j}}^{O(n)} n(\nu_i))$, which is $O(n^2)$, since the total number of children of all the rigid nodes is $O(n)$.

For each parallel component $\mu_p \in P$, by Lemmas 16 and 21, $t(\mu_p)$ and $\hat{A}(\mu_p)$ can be computed in $O(\delta(\mu_p) \sum_{k=1}^{\delta(\mu_p)} \sum_{i=1}^{k} n(\nu_i))$ time. Hence, the overall complexity for all the parallel nodes is $O(\overbrace{\sum_{\mu_p \in P} \delta(\mu_p)}^{O(n^2)} \overbrace{\sum_{k=1}^{\delta(\mu_p)} \sum_{i=1}^{k} n(\nu_i)}^{O(n)})$, which is $O(n^3)$, since the total number of children of all the parallel nodes is $O(n)$. The time complexity of the bottom-up traversal is $O(n^2) + O(n^2) + O(n^3) = O(n^3)$. Starting from the gist of the admissible set of the root, the minimum depth is computed, by Lemma 22, in $O(n)$ time.

The space bound can be obtained by considering that there are $O(n)$ components in $\mathcal{T}$ and that, by Property 7, the size of the gists of their admissible sets is $O(n)$. □

To produce a minimum-depth embedding of G with an edge e on the external face we need some additional information to be added to each component during the bottom-up traversal of $\mathcal{T}$, meant to describe how the components must be attached together in order to obtain an embedding satisfying each pair of the gist of the admissible set.

Namely, for each node μ and for each pair $p \in \hat{A}(\mu)$ we attach an "embedding descriptor" composed of:

- A Boolean variable b_μ specifying whether μ must be attached to its parent component ν with f_l^μ corresponding to f_l^ν or not
- An integer pair p_i for each child component ν_i of μ specifying how ν_i must be, in its turn, embedded in order to obtain an embedding of μ satisfying p
- If μ is a parallel component we also record the needed ordering of its child components ν_i

The minimum-depth embedding is computed with a top-down traversal of the SPQR-tree $\mathcal{T}$ rooted at e, using the above described additional structures, by suitably replacing each virtual edge with the skeleton of the corresponding component.

Theorem 2 *Let G be an n-vertex biconnected planar graph. A minimum-depth embedding of G can be computed in $O(n^4)$ time and $O(n^3)$ space.*

Proof: For each edge e of G, compute the SPQR-tree rooted at e in $O(n)$ time and the minimum-depth embedding with e on the external face in $O(n^3)$ time. The cubic space bound is due to the fact that, for each component and for each integer pair of the gist of its admissible set, an integer pair for each children must be recorded. □

5 Extension to General Planar Graphs

The minimum-depth embedding of a simply-connected planar graph G, described by its BC-tree, can be found with an approach similar to that used in [3]. The key point of such an approach is that the algorithm to compute a minimum-depth embedding of a biconnected graph with a specified edge on the external face can be suitably modified in order to be applied to each block μ_i, taking into account the depth of the blocks that are attached to the cut-vertices of μ_i, and maintaining the $O(n_i^3)$ time complexity, where n_i is the number of vertices of μ_i.

Each child block μ_j, sharing the cut-vertex v_j with μ_i, will be embedded with v_j on its external face. Hence, we apply the modified algorithm using as reference edge each one of the edges incident to v_j and choose the embedding with minimum depth.

We start by choosing a root block for the BC-tree, and a reference edge inside such a block. We traverse bottom-up the BC-tree applying the modified algorithm. This computation has to be performed for each edge of each block chosen as the root block. The overall $O(n^4)$ complexity can be obtained by considering that the modified algorithm has to be launched at most three times for each edge of G. Namely, we launch the algorithm on each edge e of G when such an edge is chosen to be on the external face, taking into account the depths of all the attached blocks, and we launch the algorithm on each edge e incident to a cut-vertex v (hence, at most two times for each e) taking into account the depths of all the attached blocks with the exception of those attached to v. Therefore, the following theorem follows.

Theorem 3 *Let G be an n-vertex connected planar graph. A minimum-depth embedding of G can be computed in $O(n^4)$ time and $O(n^3)$ space.*

Here we provide a sketch of the algorithm to find a minimum-depth embedding of a block μ of a BC-tree $\mathcal{B}$ with a specified edge e on the external face, suitably modified in order to take into account the depths of the child blocks ν_i sharing a cut-vertex v_i with μ.

The input of the modified algorithm is a minimum-depth embedding for each child block ν_i of μ with v_i on the external face. The algorithm described in Section 4 for computing a minimum-depth embedding of a biconnected graph with a specified edge on the external face has to be modified in order to take into account the fact that each child block ν_i has to be placed inside one of the faces of μ incident to v_i.

Consider the skeleton of a series, rigid, or parallel component ρ of the SPQR-tree of the block μ. The child blocks of μ, whose depths have to be taken into account, may only attach to the poles of the child components of ρ that are not poles of ρ. In fact, child blocks attached to the poles of ρ will be taken into account when the parent of ρ is considered by the algorithm. Hence, the computation for a parallel component ρ_p is unchanged with respect of the algorithm described in 4.1.

The key observation for modifying the algorithm for a series or a rigid component ρ is that the admissible set of $sk(\rho)$ and the the admissible sets of each child component ρ_i nested into $\Gamma^j(\rho)$ do not change whichever will be the faces of $sk(\rho)$ that will be chosen to contain the child blocks ν_i attached to μ.

The second key observation is that each block ν_i, with depth d_i, can be modeled by means of an additional admissible set representing the contribution of the block on the depth of the whole component. Namely, for a series component ρ_s, it is sufficient to intersect the gists of the admissible sets $\hat{A}(\rho_h|\Gamma^j(\rho))$ computed for each child component ρ_h of ρ with the additional admissible set containing the pairs $\langle 0, d_i \rangle$ and $\langle d_i, 0 \rangle$, which represent the fact that the block ν_i will be placed into $f_r^{\rho_s}$ or $f_l^{\rho_s}$, respectively.

Analogously, the algorithm for a rigid component ρ_r must be modified by considering for each child block ν_i, with depth d_i, an additional admissible set representing the fact that ν_i can be placed in any face incident to v_i. Such an admissible set contains the two pairs $\langle 0, d_i + d_r(f) \rangle$ and $\langle d_i + d_l(f), 0 \rangle$ for each face f incident to v_i (its gist will contain two pairs only).

It can be proved that the above changes do not affect the overall $O(n^3)$ complexity of the algorithm, where n is the number of vertices of μ.

6 Conclusions

We presented an $O(n^3)$-time algorithm for computing a minimum-depth embedding of a biconnected planar graph with a given edge on the external face. Then, we exploited such result to solve the problem on a general planar graph in $O(n^4)$ time.

Since our approach is inspired by that in [3], it is useful to stress the similarities ad the differences between the two contributions. We take from [3] the fundamental idea of decomposing the graph into components and to separately consider each component. Also, the concept of *thickness* is the same as in [3]. In both papers there is the idea of equipping each component with pairs of integers, representing their distance from the external face. However, in [3] a pair represents the result of a "probe" that says that a certain component is feasible with that depth. In our case a set of pairs represents implicitly all the admissible values of depth of the component. The combinatorial structure of such pairs and their nice computational properties are a key ingredient of our paper. The techniques for combining the components used in the two papers are similar. However, in the critical problem of dealing with parallel compositions we develop an approach that has many new features.

The natural problem that remains open is to fill the gap from our $O(n^4)$ time to the linear time obtained in [8] for a simplified version of the problem.

References

[1] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. of the Ass. for Comp. Mach.*, 41:153–180, 1994.

[2] D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM-J. on Comp.*, 17:53–76, 1988.

[3] D. Bienstock and C. L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990.

[4] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM J.C.*, 25(5):956–997, 1996.

[5] E. Di Giacomo, W. Didimo, G. Liotta, and H. Meijer. Computing radial drawings on the minimum number of circles. In *Graph Drawing, New York, 2004*, pages pp. 250–261, 2004.

[6] D. Dolev, F. T. Leighton, and H. Trickey. Planar embedding of planar graphs. *Adv. in Comp. Res.*, 2, 1984.

[7] M. Pizzonia. Minimum depth graph embeddings and quality of the drawings: An experimental analysis. In *Proc. GD '05*, volume 3843 of *L.N.C.S.*, pages 397–408, 2005.

[8] M. Pizzonia and R. Tamassia. Minimum depth graph embedding. In *Proc. ESA 2000*, volume 1879 of *L.N.C.S.*, pages 356–367, 2000.

[9] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.