Determining the smallest k such that G is k-outerplanar

Frank Kammer

Institut für Informatik, Universität Augsburg, 86135 Augsburg, Germany
kammer@informatik.uni-augsburg.de

Abstract. The outerplanarity index of a planar graph G is the smallest k such that G has a k-outerplanar embedding. We show how to compute the outerplanarity index of an n-vertex planar graph in $O(n^2)$ time, improving the previous best bound of $O(k^3 n^2)$. Using simple variations of the computation we can determine the radius of a planar graph in $O(n^2)$ time and its depth in $O(n^3)$ time.
We also give a linear-time 4-approximation algorithm for the outerplanarity index and show how it can be used to solve maximum independent set and several other NP-hard problems faster on planar graphs with outerplanarity index within a constant factor of their treewidth.

Key words: outerplanarity index, k-outerplanar, fixed-parameter algorithms, NP-hard, SPQR trees

1 Introduction

A promising approach to solve NP-hard graph problems to optimality is to deal with fixed-parameter algorithms, i.e., the aim is to solve NP-hard problems on an n-vertex graph in $O(f(k) \cdot n^{O(1)})$ time for some function f that depends only on some parameter k but not on n. For example, we can use the so-called *treewidth* as the parameter k. The treewidth of a graph G measures the minimum width of a so-called *tree decomposition* of G and, intuitively, describes how treelike G is. Tree decompositions and treewidth were introduced by Robertson and Seymour [15]. Given a tree decomposition of width k of an n-vertex graph G, one can solve many NP-hard problems such as maximum 3-coloring, maximum independent set, maximum triangle matching, minimum edge dominating set, minimum dominating set, minimum maximal matching and minimum vertex cover on G in $O(c^k n)$ time for some constant c. For a definition of these problems see, e.g., [1] and [8].

Many algorithms for computing the treewidth have been published. Reed [13] showed that a tree decomposition of width $O(k)$ can be found for an n-vertex graph G of treewidth k in $O(3^{3k} k \cdot n \log n)$ time. Thus, many NP-hard problems can be solved on G in $O(c^k n \log n)$ time for a constant c. Moreover, Bodlaender [4] gave an algorithm that computes a tree decomposition of width k of an n-vertex graph G of treewidth k in $\Theta(f(k)n)$ time for some exponential function f. The author only states that $f(k)$ is very large; however, Röhrig [14] shows that

$f = 2^{\Theta(k^3)}$. Much research for finding tree decompositions has also considered special classes of graphs, e.g., the (k-outer)planar graphs.

Definition 1. *An embedding φ of a planar graph is 1-outerplanar if it is outerplanar, i.e., all vertices are incident on the outer face in φ. An embedding of a planar graph is k-outerplanar if removing all vertices on the outer face (together with their incident edges) yields a $(k-1)$-outerplanar embedding.*

A graph is k-outerplanar *if it has a k-outerplanar embedding. The* outerplanarity index *of a graph G is the smallest k such that G is k-outerplanar.*

Using the ratcatcher algorithm of Seymour and Thomas [17] and the results of Gu and Tamaki [9], one can obtain a tree decomposition of width $O(k)$ for a k-outerplanar graph in $O(n^3)$ time. All of the NP-hard problems mentioned above remain NP-hard even on planar graphs [8]. In the special case of planar graphs, the outerplanarity index is a very natural parameter to use for a fixed-parameter algorithm. Bienstock and Monma [2] showed that for a planar n-vertex graph, the outerplanarity index k and a k-outerplanar embedding can be found in $O(k^3 n^2)$ time. A general technique due to Baker [1] enables us to solve each of the NP-hard problems mentioned above on k-outerplanar n-vertex graphs G in $O(c^k n)$ time for a constant c if a k-outerplanar embedding of G is given.

Using a new and simple algorithm presented here, we can find the outerplanarity index k and a k-outerplanar embedding in $O(n^2)$ time. Moreover, a slightly modified version of the new algorithm is 4-approximative and runs in linear time. Using the approximation algorithm and Baker's technique, we can solve many NP-hard problems to optimality on k-outerplanar graphs in $O(c^k n)$ time for some constant c (e.g., maximum independent set in $O(8^{4k} n)$ time and maximum triangle matching in $O(16^{4k} n)$ time). Thus, this approach is the fastest for many NP-hard problems for planar graphs whose outerplanarity index k is within a constant factor of their treewidth. Moreover, given a k-outerplanar graph, using the new algorithms one can find tree decompositions of width $3k-1$ and $12k-4$ in $O(n^2)$ time and in $O(kn)$ time, respectively [5, 16].

In the following we will consider only the outerplanarity index. However our approach can also be used to determine other distance measures such as the radius and the width. For a definition of these distance measures, see [2].

2 Ideas of the Algorithm

An often used technique is to remove some subgraph C from a given graph G and solve a problem recursively on the remaining graph G'. (G' is later called the *induced graph* of G and C.) Unfortunately, the outerplanarity index of G is not a function of the outerplanarity indices of G' and C alone. However, one can determine the outerplanarity index of G by a computation of the so-called *weighted outerplanarity index* of C—this is a small generalization of the outerplanarity index—that additionally takes into account the weighted outerplanarity indices of $\ell \in \mathbb{N}$ different graphs, each of which is G' with a few additional edges. It turns out that by a recursive call one can compute the weighted outerplanarity indices of the ℓ graphs simultaneously. The exact value of ℓ depends on the distance measure. For the (weighted) outerplanarity index and the radius, ℓ is 6, and for the width, ℓ is the number of edges. One can observe that this leads to time bounds of $O(n^2)$ for determining the radius of an n-vertex planar graph and $O(n^3)$ time for determining its depth.

As we observe later, the computation of the weighted outerplanarity index is a slight modification of the computation of the outerplanarity index. For the time being let us consider only the outerplanarity index. For a fast and simple computation of the outerplanarity index of C, we have to choose C in a special way. Before going into the details, we need additional terminology. A *rooted embedding* of a planar graph G is a combinatorial embedding of G with a specified outer face. An *embedded graph* (G, φ) is a planar graph G with a rooted embedding φ. A graph G is *biconnected* (*triconnected*) if no removal of one vertex (two vertices) from G disconnects G. As an auxiliary tool for describing the computation of the outerplanarity index, we consider so-called *peelings*. These are defined precisely in the next section. For now it will suffice to think of a peeling as a process that removes the vertices of a graph in successive steps, each of which removes all vertices incident on the outer face. Define the *peeling index* of an embedded graph (G, φ) as the minimal number of steps to remove all vertices of G. Let us call an embedding *optimal* (*c-approximative*) if it is a rooted embedding φ of a planar graph G such that the peeling index of (G, φ) equals (is bounded by c times) the outerplanarity index of G.

Suppose that C is a triconnected graph. A theorem of Whitney [19] states that the combinatorial embedding of C is unique. Moreover, it can be found in linear time [3, 12]. Hence, if one face f is chosen as the outer face, the rooted embedding φ of C is also unique. Obviously, we can determine the peeling index k of (C, φ) in linear time. Let φ^{OPT} be an optimal embedding of C. Define f^{OPT} as the outer face of φ^{OPT} and k^{OPT} as the peeling index of $(C, \varphi^{\mathrm{OPT}})$. If the *distance* of two faces f_1 and f_2 is taken to be the minimal number of edges we have to cross by going from f_1 to f_2, the distance in φ^{OPT} from f to f^{OPT} and from f^{OPT} to any other face is at most k^{OPT}. Consequently, k is at most $2k^{\mathrm{OPT}}$. By iterating over all ($\leq 2n$) faces of a combinatorial embedding of G, we find an optimal embedding. Therefore we can conclude the following.

Corollary 2. *Given a triconnected graph G, a 2-approximative and an optimal embedding of G can be obtained in linear and in quadratic time, respectively.*

3 Peelings and Induced Graphs

A *separation vertex* (*separation pair*) of a graph G is a vertex (a pair of vertices) whose removal from G disconnects G. A *biconnected component* of G is a maximal biconnected subgraph of G. Say (G, φ) is biconnected and triconnected, respectively, if G is. Call a function a *weight function* for G if it maps each vertex and each edge of G to a non-negative rational number. Let us call a vertex or edge *outside* in a rooted embedding φ if it is incident on the outer face of φ. For an edge e in an embedded graph (G, φ), define $\varphi - e$ as the embedding obtained by removing e and merging the two faces f_1 and f_2 incident on e in φ to a face f_3 in $\varphi - e$. Then f_3 *contains* the faces f_1 and f_2. Let $G = (V_G, E_G)$ be a connected embedded graph with weight function r and let φ be a rooted embedding of G. Define $Out(\varphi)$ as the set of outside vertices. If $\mathcal{S} = Out(\varphi)$ and $u, v \in \mathcal{S}$, $\mathcal{S}^{\varphi}_{u \to v} \subset \mathcal{S}$ is the set of vertices visited on a shortest clockwise travel around the outer face of φ from u to v, including u and v. If $u = v$, $\mathcal{S}^{\varphi}_{u \to v} = \{u\}$. Given a directed edge (u, v), we say that an embedding φ' is obtained from φ by *adding* (u, v) *clockwise if* φ' is a (planar) rooted embedding of $(V_G, E_G \cup \{\{u, v\}\})$ such that all inner faces of φ are also inner faces of φ' and $Out(\varphi') = Out(\varphi)^{\varphi}_{v \to u}$. By iterating over a set of directed edges E^+ we can *add* E^+ *clockwise* into an embedded graph. The embedding obtained does not depend on the order in which edges are added since the edges in E^+ can be added clockwise only if they do not interlace. Let E^+ be a set of directed edges that all can be added clockwise into φ. Call the resulting embedding φ^+. The *peeling* $\mathcal{P}$ of the quadruple (G, φ, r, E^+) is the (unique) list of vertex sets $(V_1, \ldots, V_t)$ with $\bigcup_{i=1}^{t} V_i = V_G$ and $V_t \neq \emptyset$ such that each set V_i $(1 \leq i \leq t)$ contains all vertices that are incident on the outer face obtained after the removal of the vertices in $V_1, \ldots, V_{i-1}$ in the *initial embedding* φ^+. Thus, *(peeling) step* i removes the vertices in V_i. Note that after each step i we obtain a unique embedding of the subgraph $G[V \setminus \bigcup_{j=1}^{i} V_j]$. Define the *peeling number* of a vertex $u \in V_i$ and an edge $\{u, v\}$ with $v \in V_j$ $(1 \leq i, j \leq t)$ in the peeling $\mathcal{P}$ as $\mathcal{N}_{\mathcal{P}}(v) = i + r(v)$, $\mathcal{N}_{\mathcal{P}}(\{u, v\}) = \min\{i, j\} + r(\{u, v\})$ and $\mathcal{N}_{\mathcal{P}}(G) = \max\{\max_{v \in V_G} \mathcal{N}_{\mathcal{P}}(v), \max_{e \in E_G} \mathcal{N}_{\mathcal{P}}(e)\}$. For simplicity, if r is clear from the context, let a peeling of (G, φ) be the peeling of $(G, \varphi, r, \emptyset)$.

The *weighted outerplanarity index* of G with weight function r is the minimal peeling number of the peelings of $(G, \varphi', r, \emptyset)$ over all rooted embeddings φ' of G. Note that the outerplanarity index of G is the weighted outerplanarity index in the special case $r \equiv 0$.

For a subgraph $H = (V_H, E_H)$ of G, let us say that two vertices u and v are connected in G by an *internally H-avoiding path* if there is a path from u to v in G, no edge of which belongs to H. Moreover, an *H-attached component* in G is a maximal vertex-induced subgraph $G' = G[V']$ of G such that each pair of vertices in V' is connected in G by an internally H-avoiding path. A subgraph H of G is a *peninsula* of G if $H = G$ or each H-attached component of G has at most two vertices in common with H. An *outer component* C of a peninsula H of G is defined in φ as H itself or as an H-attached component of G such that C contains an outside edge in φ. Next we define the induced graph $H(G)$ of a peninsula H of G and its embedding $\varphi_{H(G)}$ inherited from of the embedding φ of G. For an example, see Fig. 1. The motivation for this definition is that—as we observe later—peelings of (G, φ) and of $(H(G), \varphi_{H(G)})$ are very similar.

Definition 3 (Induced graph). *Given a connected planar graph G and a peninsula $H = (V_H, E_H)$ of G, the* induced graph $H(G)$ *has the vertex set V_H and the edge set $\{\{u, v\} \mid (\{u, v\} \in E_H) \vee (u, v \in V_H$ and u and v are connected in G by an internally H-avoiding path $)\ \}$. An edge $\{u, v\}$ in $H(G)$ is called a* virtual edge *if there is an internally H-avoiding path between u and v in G (even if $\{u, v\} \in E_H$).*

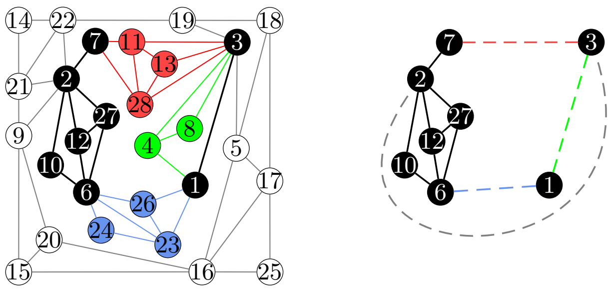

Fig. 1. Left: An embedded graph (G, φ) with a peninsula H (black), an outer component of H in G (all white vertices and vertices 2 and 3) and three further H-attached components in G. Right: The inherited embedding $\varphi_{H(G)}$ with the virtual edges dashed.

A *triconnected component* of G is the triconnected induced graph $H(G)$ of a peninsula H of G such that there exists no subgraph H' of G for which H is a subgraph of H' and $H'(G)$ is triconnected. For each peninsula H, define $\mathcal{R}_H$ as the function that maps each H-attached component C to the set of vertices common to C and H. Let C be a H-attached component. By definition of a peninsula, $|\mathcal{R}_H(C)| \leq 2$. If $|\mathcal{R}_H(C)| = 2$ we usually take $\mathcal{R}_H(C)$ as a virtual edge instead of a set of two vertices.

Definition 4 (Inherited embedding). *Given a connected embedded graph* (G, φ) *and a peninsula* H *of* G*, an* embedding φ' of H inherited from φ *is a rooted embedding of* H *for which the edges incident on each vertex* v *of* H *appear around* v *in the same order in* φ *and in* φ'*. Moreover, the outer face of* φ' *contains the outer face of* φ*.*

For the induced graph $H(G)$ *of* H*, an* inherited embedding *is a rooted embedding* $H(G)$ *defined almost as before; the only difference is that each virtual edge* $\{u, v\}$ *is mapped to the cyclic position around* u *of an edge incident on* u *of a simple internally* H*-avoiding path from* u *to* v *in* G*.*

Observe that the inherited embedding of an induced peninsula is unique if at each vertex u, a virtual edge $\{u, v\}$ can be mapped only to a set of consecutive edges around u, none of which is part of the peninsula. Let us call a peninsula H *good* if this is the case and if additionally for each virtual edge $\{u, v\}$ only one H-attached component C exists such that $\mathcal{R}_H(C) = \{u, v\}$. The last condition allows us later to identify each H-attached component with a virtual edge. For each embedded graph (G, φ) and for each a good peninsula H of G, define the *embedding* φ_H as the embedding of H inherited from φ. In the rest of this section we consider different properties of peelings and of inherited embeddings.

Lemma 5. *Let* (G, φ) *be a connected embedded graph and let* e *be an edge of* G *incident on the outer face in* φ*. For each good peninsula* H *of* G*, either* e *is in* H *and incident on the outer face in* $\varphi_{H(G)}$*, or* e *is in some* H*-attached component* C *in* G *and* $\mathcal{R}_H(C)$ *is incident on the outer face in* $\varphi_{H(G)}$*.*

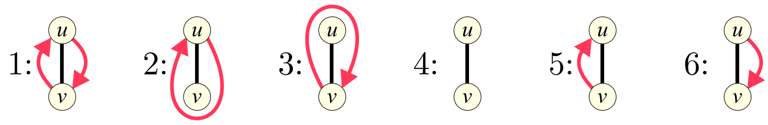

Fig. 2. Possibilities for an edge to be outside, e.g., 1: Both endpoints are outside.

Proof. If e is in H, i.e., e is in $H(G)$, e remains outside. Otherwise, e is in $C \neq H$. Let $\{u, v\} = \mathcal{R}_H(C)$—possibly $u = v$. Let p a internally H-avoiding path from u to v using e such that p contains no simple cycle. The faces of φ are merged in $\varphi_{H(G)}$ such that at least one side of p is outside in $\varphi_{H(G)}$.

Given an embedded graph (G, φ), the *enclosure* of a vertex set S in G is the maximal subgraph C of G such that the only vertices outside in φ_C are the vertices of S.

Observation 6 *Let (G, φ) and (G', φ') be embedded graphs and let S and S' be vertex sets in these graphs, respectively, such that between the enclosure S^+_G of S in (G, φ) and the enclosure $S^+_{G'}$ of S' in (G', φ') an graph isomorphism f exists such that vertices u and v from S^+_G are adjacent if and only if $f(u)$ and $f(v)$ are adjacent in $S^+_{G'}$. Moreover, let r and r' be weight functions for G and G', respectively, such that $r(v) = r'(f(v))$ for all vertices v from S^+_G. If $\mathcal{N}_{\mathcal{P}}(v) - \mathcal{N}_{\mathcal{P}'}(f(v))$ is constant on all $v \in S$, then it is constant on all $v \in S^+_G$.*

Let (G, φ) be a connected embedded graph with weight function r and let $H = (V_H, E_H)$ be a good peninsula with an edge $\{u^*, v^*\}$ outside in φ. Informally, we now want to compare the peeling of (G, φ) with the peeling of the inherited embedding of $(H(G), \varphi_{H(G)})$. Moreover, what happens with an H-attached component $C = (V_C, E_C)$ of G during the peeling of φ?

Define $\mathcal{L}(V^+)$ for a set $V^+ = \{u\}$ as the list of edge sets $(\{(u, u)\}, \{\})$ and for a set $V^+ = \{u, v\}$ as $(\{(u, v), (v, u)\}, \{(u, u)\}, \{(v, v)\}, \{\}, \{(v, u)\}, \{(u, v)\})$. The directed edges of Fig. 2 shows $\mathcal{L}(\{u, v\})$. Take E^+ as a set in the list $\mathcal{L}(\{u^*, v^*\})$ and $\mathcal{P} = (V_1, V_2, \ldots, V_t)$ as a peeling of (G, φ, r, E^+). Choose u and v such that $\{u, v\} = \mathcal{R}_H(C)$—possibly $u = v$. If $u \in V_i$ and $v \in V_j$ $(1 \leq i, j \leq t)$, take $q = \min\{i, j\}$. For the embedding φ_C of C inherited from φ, define $\mathcal{S} = Out(\varphi_C)$. Let H' be the outer component of C that is a supergraph of H and that contains $\{u^*, v^*\}$. Let $\mathcal{P}'$ be the peeling of $(H'(G), \varphi_{H'(G)}, r', E^+)$ where r' is an arbitrary weight function such that r' is equal to r for all vertices of $H'(G)$. The following two properties are proved below.

Property 1. $V_q \cap V_C$ equals one of the 6 sets below. In other words, if we ignore the vertices not in C, peeling step q removes one of these 6 sets: 1. $\{u, v\}$, 2. $\{u\}$, 3. $\{v\}$, 4. $\mathcal{S}$, 5. $\mathcal{S}^{\varphi_C}_{u \to v}$ or 6. $\mathcal{S}^{\varphi_C}_{v \to u}$. The remaining vertices of $\mathcal{S}$ are removed in step $q + 1$.

Property 2. The removal of C from G if $|\mathcal{R}_H(C)| = 1$ and the replacement of C by the (virtual) edge $\mathcal{R}_{H'}(C)$ in G if $|\mathcal{R}_H(C)| = 2$ does not change the peeling numbers of the vertices in H', i.e., $\mathcal{N}_{\mathcal{P}}(H') = \mathcal{N}_{\mathcal{P}'}(H')$.

For each of the 6 sets of Property 1 an example in Fig. 2 shows a situation just before the *qth* peeling step. In each example the black edge corresponds to C and each directed edge has a corresponding undirected path in H.

Proof of Property 1. Consider the two cases $|\mathcal{R}_H(C)| = 1$ and $|\mathcal{R}_H(C)| = 2$. For the first case (i.e., $u = v$), observe that if one vertex of $\mathcal{S} \setminus \{u\}$ is removed in peeling step q, then all vertices in $\mathcal{S}$ are removed in that step because there is no edge from $\mathcal{S} \setminus \{u\}$ to H. Thus, $V_q \cap V_C$ equals $\{u\}$ or $\mathcal{S}$. In the second case we argue in the same way that if a vertex in $\mathcal{S}^{\varphi_C}_{u \to v} \setminus \{u, v\}$ or $\mathcal{S}^{\varphi_C}_{v \to u} \setminus \{u, v\}$ is removed, then all vertices in $\mathcal{S}^{\varphi_C}_{u \to v}$ or $\mathcal{S}^{\varphi_C}_{v \to u}$, respectively, are removed. The remaining vertices in $\mathcal{S}$ are outside after step q.

Proof of Property 2. Starting with the initial embedding $\varphi_0^{\mathcal{P}} = \varphi$ and $\varphi_0^{\mathcal{P}'} = \varphi_{H'(G)}$ and carrying out in parallel the peelings $\mathcal{P}$ and $\mathcal{P}'$, let us compare the embeddings $\varphi_i^{\mathcal{P}}$ and $\varphi_i^{\mathcal{P}'}$ obtained after $i = 1, 2, \ldots$ peeling steps. By induction on i, we can observe that if a *C-internal face* is one that is incident only on vertices of C, there is the following bijection between the not C-internal faces of $\varphi_i^{\mathcal{P}}$ and the faces of $\varphi_i^{\mathcal{P}'}$: Each face f in $\varphi_i^{\mathcal{P}}$ is mapped to the face in $\varphi_i^{\mathcal{P}'}$ whose boundary vertices are those of f, except for the vertices in C. Thus, we can conclude that a face of $\varphi_i^{\mathcal{P}}$ is the outer face if and only if the corresponding face of $\varphi_i^{\mathcal{P}'}$ is the outer face, i.e., a vertex in H' is outside after the same number of peeling steps in $\mathcal{P}$ and in $\mathcal{P}'$.

Since C is the enclosure of $\mathcal{S}$, we can apply Observation 6. Informally, the peeling numbers of vertices in C depend only on the peeling numbers of vertices in $\mathcal{S}$. Because of that and Property 1, a set of edges $E' \in \mathcal{L}(\{u, v\})$ exists such that for the peeling $\mathcal{P}''$ of (C, φ_C, r, E') and all $v \in C$: $\mathcal{N}_{\mathcal{P}}(v) = \mathcal{N}_{\mathcal{P}''}(v) + q - 1$. Observe that E' can be determined during the peeling $\mathcal{P}'$ of $(H'(G), \varphi_{H'(G)}, r', E^+)$ by observing which parts of the virtual edge $\mathcal{R}_H(C)$ are outside just before it is removed. For that reason, let us define E' as the *induced set of extra edges* of $\mathcal{P}'$ and $\mathcal{R}_H(C)$. Figure 2 shows various possibilities for how a virtual edge (black) can be outside.

The removal and replacement described in Property 2, applied to all H-attached components of G in φ, results (after the removal of multiple edges) in the inherited embedding $\varphi_{H(G)}$. Thus, for the vertices in H, the peeling numbers in the peeling of (G, φ, r, E^+) and in the peeling $\mathcal{P}^*$ of $(H(G), \varphi_{H(G)}, r', E^+)$ coincide for each set E^+ where r' is—except to the following two changes—equal to r. The changes enables us to add the peeling numbers of all H-attached components to the computation of the peeling $\mathcal{P}^*$. The first change is to extend the domain of the weight function r to all virtual edges $\mathcal{R}_H(C')$ and set $r(\mathcal{R}_H(C'))$ to -1 plus the peeling number of C' in the peeling $(C', \varphi_{C'}, r, E')$ where E' is the *induced set of extra edges* of $\mathcal{P}^*$ and $\mathcal{R}_H(C')$ and the second change is to increase $r(v)$ by -1 plus the highest peeling number of an H-attached component C^* with $\mathcal{R}_H(C^*) = \{v\}$ $(v \in V_H)$. Leave $r(v)$ unchanged if no such C^* exists. Denote the function obtained the *$(G, H, \mathcal{P}^*)$-extended and (G, H)-increased weight function* of r.

Given a graph G, a set of directed edges E' and a set S of vertices (edges), an embedding φ is *vertex-constrained* (*edge-constrained*) *optimal* for (G, r, E', S)

if all $s \in S$ are outside with respect to φ and for all embeddings φ' the following is true: Either not all $s \in S$ are outside with respect to φ' or $\mathcal{N}_{\mathcal{P}'}(G) \leq \mathcal{N}_{\mathcal{P}''}(G)$, where $\mathcal{P}'$ and $\mathcal{P}''$ are the peelings of (G, φ, r, E') and (G, φ', r, E'), respectively. Note that a smallest peeling number and a rooted embedding with this peeling number can be computed simultaneously.

Theorem 7. *Given a graph $G = (V_G, E_G)$ with a weight function r and an edge $\{u^*, v^*\} \in E_G$, one can find an edge-constrained optimal embedding of $(G, r, E^+, \{u^*, v^*\})$ for all $E^+ \in \mathcal{L}(\{u^*, v^*\})$ as follows:*

Taking a good peninsula $H = (V_H, E_H)$ of G that contains $\{u^*, v^*\}$, first determine for each H-attached component C the constrained optimal embedding of $(C, r, E', \mathcal{R}_H(C))$ for all $E' \in \mathcal{L}(\mathcal{R}_H(C))$ recursively.

For each E^+ and each embedding φ of $H(G)$ with $\{u^*, v^*\}$ outside, then determine the peeling number of the peeling $\mathcal{P}$ of $(H(G), \varphi, r', E^+)$, where r' is the $(G, H, \mathcal{P})$-extended and (G, H)-increased weight function of r.

For each E^+ finally return the smallest peeling number and a corresponding embedding φ.

4 The Outerplanarity Index for Biconnected Graphs

Before applying Theorem 7, we have to decompose G into good peninsulas. Given a biconnected planar graph G, this can be done in linear time by constructing a so-called SPQR tree of G [11]. An *SPQR tree* [6] of a biconnected graph G is a tree $\mathcal{T} = (\mathcal{W}, \mathcal{F})$ with a mapping $\mathcal{M}$ from $\mathcal{W}$ to a set of so-called *split-components* of G. Each split-component—consisting of either a *triconnected*, a *cycle* or a *multiple-edge component* as defined later—is the induced graph $H(G)$ of a good peninsula H of G, which, in the case of a multiple-edge component, is extended by some edges. Additionally, for an SPQR tree the following two properties hold: First, each edge of G is part of exactly one split-component $H \in \mathcal{M}(\mathcal{W})$ and second, two nodes $w_1, w_2 \in \mathcal{W}$ with $H_i = \mathcal{M}(w_i)$ $(i = 1, 2)$ are connected by an edge in $\mathcal{F}$ if and only if $\mathcal{R}_{H_1}(H_2) \neq \emptyset$. A cycle component is a simple cycle and a multiple-edge component consists of only 2 vertices (a separation pair of G) connected by at most one edge of G and several virtual edges. In addition, let us extend the mapping $\mathcal{M}$ from the nodes of $\mathcal{T}$ to each subtree T' of $\mathcal{T}$ in the canonical way. In detail, if we take $\{w_1, \ldots, w_\ell\}$ as the nodes of T', $(W_i, F_i) = \mathcal{M}(w_i)$ and S as the set of all virtual edges $\{e \mid e \in F_i \cap F_j$ and $(1 \leq i < j \leq \ell)\}$, define $\mathcal{M}(T') = (\cup_{i=1}^{\ell} W_i, \cup_{i=1}^{\ell} F_i \setminus S)$.

For a graph G and an edge e of G, a *rooted SPQR tree* $\mathcal{T} = (\mathcal{W}, \mathcal{F})$ of (G, e) is an SPQR tree of G rooted at the unique node $r \in W$ with $e \in \mathcal{M}(r)$. The *subtree module* of a node w in $\mathcal{T}$ is the graph $\mathcal{M}(T_w)$, where T_w is the maximal subtree of $\mathcal{T}$ with root w. Observe that in a rooted SPQR tree each subtree module is the induced graph of a good peninsula. Let $w \in \mathcal{W}$ and let $C = \mathcal{M}(w)$. The *virtual parent-edge* e_p of C and of the subtree module of w is defined as follows. If w is the root of a rooted SPQR tree of (G, e), take $e_p = e$; otherwise, take $e_p = \mathcal{R}_C(H)$ with $H = \mathcal{M}(w')$ and w' being the parent of w.

Given a biconnected planar graph G, a weight function r and an edge e_{out} of G, let us now compute an edge-constrained optimal embedding of G with e_{out} outside since we can later use this computation as an auxiliary procedure for finding an optimal or an approximative embedding. Recall that a rooted embedding φ of G is optimal (c-approximative) if the peeling steps of the peeling of $(G, \varphi, r, \emptyset)$ equals (is bounded by c times) the outerplanarity index of G. Note that the peeling steps do not depend on r.

First, we compute a rooted SPQR tree $\mathcal{T}$ of (G, e_{out}) and then traverse it bottom-up. Let w be the current node visited by the traversal and let $e_p = \{u, v\}$ be the virtual parent edge of w. Define $H = \mathcal{M}(w)$, C as the subtree module of w and φ as some constrained optimal embedding of $(G, r, \emptyset, e_{\text{out}})$. In the following we determine an embedding $\varphi_{C'}$ of $C' = C - e_p$ inherited from φ. Let $H' = H - e_p$. By Lemma 5, e_p is incident on the outer face of φ_C, i.e., u and v are incident on the outer face of $\varphi_{C'}$. By Theorem 7, for all $E^+ \in \mathcal{L}(\{u, v\})$ we can determine a constrained optimal embedding $\varphi_{C'}$ of (C', r, E^+, e_p) by considering the peeling $\mathcal{P}$ of $(H', \varphi_{H'}, r', E^+)$ for all embeddings $\varphi_{H'}$ of H' with e_p outside and r' the $(C', H', \mathcal{P})$-extended weight function of r. The reason for this is that recursive calls have already found the constrained optimal embeddings for the subtree modules of all children of w.

If H is a triconnected or a cycle component, there are only two possible rooted embeddings of H with e_p outside. Thus, there is only one rooted embedding of H' with u and v outside. Therefore, $\varphi_{H'}$ and also $\varphi_{C'}$ can be found in time linear in the size of H. If $H = (\{u, v\}, E_H)$ is a multiple-edge component, both vertices of H have to be outside. Depending on E^+, we can choose $d \in \{0, 1, 2\}$ children of w such that their subtree modules are (with an edge) outside in the embedding obtained from $\varphi_{C'}$ by adding E^+ clockwise. Observe that the peeling numbers of a subtree module of a child of w differ only by at most one for different orders of the embeddings of the subtree modules between u and v. Let $C_1, \ldots, C_\ell$ be the subtree modules of the children of w. The *inside peeling number* of C_i ($i \in \{1, \ldots, \ell\}$) is the number of peeling steps in a constrained optimal embedding of $(C_i, r, \{(u, v), (v, u)\}, \{u, v\})$ and the *outside peeling number* of C_i is the smaller number of peeling steps in the constrained optimal embeddings of $(C_i, r, \{(u, v)\}, \{u, v\})$ and of $(C_i, r, \{(v, u)\}, \{u, v\})$. Since the peeling number of C' is the maximum over the peeling numbers of all C_i ($1 \le i \le \ell$), our goal is to reduce the largest peeling number. Therefore, we choose for C' an embedding such that d subtree modules of the children of w with largest inside peeling number have outside vertices in $\varphi_{C'}$. Finding d subtree modules with largest inside peeling number (i.e., finding $\varphi_{C'}$) needs time linear in the size of H.

Lemma 8. *Given a biconnected graph G with a weight function r and an edge e_{out} of G, an edge-constrained optimal embedding of $(G, r, \emptyset, e_{\text{out}})$ can be found in linear time.*

For an optimal embedding of a given graph $G = (V, E)$, iterate over all edges $e \in E$ and for an approximative embedding of G consider only one arbitrary edge $e \in E$. In each iteration use weight function $r \equiv 0$ and compute an edge-constrained optimal embedding with e outside. The quality of the approximative embedding φ with edge e outside can be estimated with arguments similar to those used in the proof of Corollary 2. Let φ^{OPT} be an optimal embedding of G and let f^{OPT} be the outer face of φ^{OPT}. Choose φ' as the rooted embedding such that φ^{OPT} and φ' have the same combinatorial embedding and e is outside. The number of peeling steps to remove (G, φ') is at most twice the number of peeling steps to remove $(G, \varphi^{\mathrm{OPT}})$. Since φ is an edge-constrained optimal embedding with $r \equiv 0$ and e outside, (G, φ) does not need more peeling steps to remove than (G, φ').

Corollary 9. *Given a biconnected planar graph G, a 2-approximative and an optimal embedding of G can be found in linear and in quadratic time, respectively.*

5 The Outerplanarity Index for General Graphs

Given a planar graph $G = (V_G, E_G)$ with weight function r, we can determine the peeling number for each connected component separately. Thus, let us assume in the following that all graphs considered are connected.

Let $e \in E_G$ and let B be the biconnected component containing e. For the moment, let us assume that we know the peeling number of a vertex-constrained optimal embedding $(H, r, \emptyset, \mathcal{R}_H(B))$ of each B-attached component H. By Theorem 7 we can then determine the (G, B)-increased weight function r' of r and use the algorithm for edge-constrained optimal embeddings on B with weight function r' to obtain an edge-constrained optimal embedding for the whole graph G with e outside. By an iteration over all edges of G an optimal embedding can be found.

It remains to show how all necessary vertex-constrained optimal embeddings can be found. For this purpose let us construct the block-cutpoint tree of G [7, 10]. A *block-cutpoint tree* of a connected graph G is a tree T whose nodes are the separation vertices and the biconnected components of G. Each biconnected component B is incident in T to all separation vertices in B. A block-cutpoint tree can be found in linear time [18]. A *rooted block-cutpoint tree* $T = (W, F, R)$ is a block-cutpoint tree (W, F) rooted at some biconnected component R. Given a rooted block-cutpoint tree $T = (W, F, R)$, the *subtree* (*supertree*) *c*omponent of a biconnected component B of G is the subgraph of G consisting of the biconnected component B and all biconnected components below (strictly above) B in T.

Let us traverse some rooted block-cutpoint tree $T = (W, F, R)$ of G first bottom-up and then top-down. During the bottom-up (top-down) traversal we compute at each biconnected component $B \neq R$ with parent v a vertex-constrained optimal embedding of the subtree (supertree) component of B with v outside as follows. Define during the traversals for each B in the bottom-up case B' as B and in the top-down case B' as the grandparent of B. Note that for each sub- and supertree component C, we already know the peeling numbers of each B'-attached component in C. Thus, we can determine the (C, B')-increased weight function r' of r. A vertex-constrained optimal embedding with a vertex v outside can be found by iterating over each edge e of B' incident to v and computing an edge-constrained optimal embedding $(B', r', \emptyset, e)$. In the end, we obtain for each separation vertex v a vertex-constrained optimal embeddings of all $(\{v\},\{\})$-attached components in G in quadratic time. Taking $r \equiv 0$ we obtain the following.

Theorem 10. *Given a planar graph G, an optimal embedding and the outerplanarity index of G can be found in quadratic time.*

For an approximate embedding of a graph G with weight function r the idea is so far to fix an arbitrary edge e and to search for an edge-constrained optimal embedding with e outside. Unfortunately, we cannot use the same approach as for an optimal embedding since determining a vertex-constrained optimal embedding of a single big biconnected component may take to much time. Therefore we take a rooted block-cutpoint tree T, compute only bottom-up for each biconnected component B with parent v and subtree component $C = (V_C, E_C)$ in T the two embeddings φ and φ' of B as defined below and return the smaller peeling number of the two peelings of (B, φ) and of (B, φ) and the corresponding embedding. As weight function use the (C, B)-increased weight function r' of r adapted from the peeling numbers obtained by recursive calls.

For some edge e incident on v take φ as an edge-constrained optimal embedding of C with e outside. If D is a grandchild of B such that the algorithm has recursively obtained on input D a biggest peeling number over all grandchildren and if v^* is the parent of D, φ' is taken to be a vertex-constrained optimal embedding of C with v and v^* outside. Note that φ' need not exist. We can find φ'—if it exists—by computing an edge-constrained optimal embedding of $(V_C, E_C \cup \{v, v^*\})$ with $\{v, v^*\}$ outside and then removing $\{v, v^*\}$. Use the results of the recursive calls to find φ and φ' in time linear in the size of B.

We now show by induction on the height of a vertex B in T that this bottom-up computation finds a *vertex-constrained 2-approximative embedding* φ of C with v outside, i.e., the peeling number of (C, φ) is at most twice the peeling number of (C, φ^*) for every other embedding φ^* of C with v outside.

Let k_B^{OPT}, k_C^{OPT} and k_D^{OPT} be the peeling number of a constrained optimal embedding of $(B, r, \emptyset, \{v\})$, $(C, r, \emptyset, \{v\})$ and $(D, r, \emptyset, \{v\})$, respectively, and let k_B, k_C and k_D be the peeling numbers obtained from our algorithm for inputs B, C and D, respectively. Although in φ we enforce one fixed edge incident to v to be outside, the extra peeling costs of $k_B - k_B^{\mathrm{OPT}}$ are at most 1 since after a first peeling step the outer face contains each face incident on v. If a vertex of B causes peeling number k_C, i.e., $k_C = k_B$, the embedding obtained for C is vertex-constrained 2-approximative since $k_C = k_B \le k_B^{\mathrm{OPT}} + 1 \le 2k_B^{\mathrm{OPT}} \le 2k_C^{\mathrm{OPT}}$. Otherwise, some subtree component H—possibly $H = D$—of some grandchild of B causes the peeling number k_C. Let v' be a child of B connecting B and H. Let k_H^{OPT} be the peeling number of a constrained optimal embedding of $(H, r, \emptyset, \{v\})$ and let k_H be the peeling number obtained from our algorithm for input H. Then $k_D \ge k_H$. Define $q = k_C^{\mathrm{OPT}} - k_H^{\mathrm{OPT}}$. Let us consider 3 cases:

$\mathbf{q > 0}:$ $k_C \le 2k_H^{\mathrm{OPT}} + (q+1) \le 2k_H^{\mathrm{OPT}} + 2q = 2k_C^{\mathrm{OPT}}$.
$\mathbf{q = 0 \wedge k_H < 2k_H^{OPT}}:$ $k_C \le k_H + (q+1) = (k_H+1) + q \le 2k_H^{\mathrm{OPT}} + 2q = 2k_C^{\mathrm{OPT}}$.

$\mathbf{q = 0 \wedge k_H \geq 2k_H^{OPT}}$: $k_D^{\mathrm{OPT}} \geq k_D/2 \geq k_H/2 \geq k_H^{\mathrm{OPT}} = k_C^{\mathrm{OPT}}$, so that in some optimal embedding v^* is outside.

In the first and second case φ and in the last case φ' gives us a vertex-constrained 2-approximative embedding. Similarly as in the proof of Corollary 9 we can conclude that a vertex-constrained 2-approximative embedding is a 4-approximative embedding. Taking $r \equiv 0$ we obtain the following.

Theorem 11. *Given a planar graph G, a 4-approximative embedding of G and a 4-approximation of the outerplanarity index of G can be found in linear time.*

References

1. B. S. Baker, Approximation algorithms for NP-complete problems on planar graphs, Journal of the ACM **41** (1994): 153-180.
2. D. Bienstock and C. L. Monma, On the complexity of embedding planar graphs to minimize certain distance measures, *Algorithmica* **5** (1990): 93-109.
3. N. Chiba, T. Nishizeki, S. Abe and T. Ozawa, A linear algorithm for embedding planar graphs using PQ-trees, J. Comput. System Sci. **30** (1985): 54-76.
4. H. L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. **25** (1996): 1305-1317.
5. H. L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, Theoretical Computer Science **209** (1998):1-45.
6. G. Di Battista and R. Tamassia, Incremental planarity testing. In Proc. 30th IEEE Symp. on Foundations of Computer Science (1989): 436-441.
7. T. Gallai, Elementare Relationen bezüglich der Glieder und trennenden Punkte von Graphen, Magyar Tud. Akad. Mat. Kutato Int. Kozl. **9** (1964): 235-236.
8. M. R. Garey and D. S. Johnson, Computers and Intractability, *A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, Calif., 1979.
9. Q. P. Gu and H. Tamaki, Optimal branch-decomposition of planar graphs in $O(n^3)$ time, ICALP 2005, LNCS 3580 (2005): 373-384.
10. F. Harary and G. Prins, The block-cutpoint-tree of a graph, Publicationes Mathematicae Debrecen **13** (1966): 103-107.
11. C. Gutwenger and P. Mutzel, A linear time implementation of SPQR trees, GD 2000, LNCS 1984 (2001): 77-90.
12. K. Mehlhorn and P. Mutzel, On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm, Algorithmica **16** (1996): 233-242.
13. B. Reed, Finding approximate separators and computing tree-width quickly, Proc. 24th Annual ACM Symp. on Theory of Computing (STOC 1992): 221-228.
14. H. Röhrig, Tree Decomposition: A Feasibility Study, Master's thesis, Max-Planck-Institut für Informatik in Saarbrücken (1998).
15. N. Robertson and P. D. Seymour, Graph minors. I Excluding a forest, J. Comb. Theory Series B **35** (1983): 39-61.
16. N. Robertson and P. D. Seymour, Graph minors. III Planar tree-width, J. Comb. Theory Series B **36** (1984): 49-64.
17. P. D. Seymour and R. Thomas, Call routing and the ratcatcher, Combinatorica **14**(2) (1994): 217-241.
18. R. E. Tarjan, Depth-first search and linear algorithms, SIAM J. Comp. **1** (1972): 146-160.
19. H. Whitney, Non-separable and planar graphs, Trans. Amer. Math. Soc. **34** (1932): 339-362.