Graph Embedding with Topological Cycle-Constraints*

Christoph Dornheim

Institut für Informatik
Albert-Ludwigs-Universität
D-79110 Freiburg, Germany
dornheim@informatik.uni-freiburg.de

Abstract. This paper concerns graph embedding under topological constraints. We address the problem of finding a planar embedding of a graph satisfying a set of constraints between its vertices and cycles that require embedding a given vertex inside its corresponding cycle. This problem turns out to be NP-complete. However, towards an analysis of its tractable subproblems, we develop an efficient algorithm for the special case where graphs are 2-connected and any two distinct cycles in the constraints have at most one vertex in common.

1 Introduction

In many graph drawing applications it is important to find drawings with certain geometrical or topological properties. For instance, to support the semantics of visual languages, given subgraphs should be drawn with a predefined shape. Such user-defined requirements on the graph layout are called graph drawing constraints. Due to their wide applicability, graph drawing approaches supporting them are of recent research interest [7]. So far most of these approaches, e.g. [5], focus on drawing arbitrary graphs with some sort of geometric constraints such as arithmetic restrictions on the coordinates of vertices. However, topological constraints that are less specific than geometric constraints have gained only little attention although they prove useful in expressing qualitative information about where to place the vertices.

In this paper we consider a special sort of topological constraints which we call *cycle-constraints*, each specifying a vertex and a cycle of a given graph with the meaning that the vertex must be embedded in the interior of the corresponding cycle. As an additional restriction, the drawings we are interested in are not allowed to have any crossings. Thus, we are concerned with the problem of finding a planar embedding of a graph satisfying a given set of cycle-constraints.

After recalling in Sect. 2 basic terminology for planar graphs, we provide in Sect. 3 a combinatorial characterization of graph embeddability with cycle-constraints which easily shows the problem to be in NP. Unfortunately, like many

* This work was supported by a DFG grant of the graduate school on human and machine intelligence.

graph drawing optimization problems, the problem also turns out to be NP-hard. On the other hand, this opens the search for relevant tractable subproblems. As a first approach we develop in Sect. 4 an $O(n^2)$ algorithm for the case where graphs are 2-connected and any two distinct cycles in the constraints have at most one vertex in common.

2 Preliminaries

Assuming familiarity with basic graph terminology, we recall in this section definitions and properties of planar graphs and their embeddings as presented e.g. in [3]. Unless otherwise stated, all graphs are finite, simple and undirected. The set of *vertices* and *edges* of a graph G are denoted $V(G)$ and $E(G)$, respectively. A cycle C in G is *induced* if any edge of G joining two vertices of C is already an edge in C. Moreover, C is *non-separating* if the number of components of G does not increase when deleting C from G.

A *plane graph* (V, E) consists of a finite set $V \subseteq \mathbb{R}^2$ and a finite set E of Jordan curves connecting two points of V such that any two points are connected by at most one curve and the interior of a curve contains no point of V and no point of any other curve. These points and curves are again called *vertices* and *edges*, respectively, since all graph notions can also be applied to plane graphs. If there is an isomorphism between an abstract graph G and a plane graph $\widetilde{G}$, we call $\widetilde{G}$ an *embedding* of G in the plane. A graph is *planar* if it has an embedding in the plane. In the sequel $\widetilde{G}$ always denotes an embedding of the graph G.

Every plane graph G divides $\mathbb{R}^2 \setminus G$ into disjoint open arcwise connected regions called *faces* of G; the unbounded one is the *outer face* and the bounded ones are the *inner faces*. If a cycle C of G is the boundary of some face, we call C a *facial cycle*; depending on whether C is the boundary of an inner or outer face, we call C an *inner* or *outer facial cycle*, respectively. The interior of C is denoted by *int* C and its closure by *Int* C; *ext* C and *Ext* C are defined analogously.

For characterizing graph embeddability with cycle-constraints we need the following notions and results concerning the cycle structure of graphs (see [8], [3]). Given a graph $G = (V, E)$ the *edge space* $\mathcal{E}(G)$ of G is the vector space over the 2-element field $\mathbb{F}_2 = \{0, 1\}$ formed by the set of all subsets of E and the sum $A \oplus B = (A \setminus B) \cup (B \setminus A)$ for $A, B \subseteq E$. The *cycle space* $\mathcal{C}(G)$ is the subspace of $\mathcal{E}(G)$ generated by the cycles of G – more precisely, by their edge sets. The elements of $\mathcal{C}(G)$ are unions of edge-disjoint cycles of G. Suppose $\widetilde{G}$ is an embedding of a 2-connected planar graph G and $\mathcal{F}'$ is the set of all facial cycles of $\widetilde{G}$. Then $\mathcal{F} = \mathcal{F}' \setminus \{F\}$ forms a basis of $\mathcal{C}(G)$ for any $F \in \mathcal{F}'$ and F is the sum of all cycles in $\mathcal{F}$. There is always an embedding of G where $\mathcal{F}$ is the set of inner facial cycles and F the outer facial cycle. Moreover, any two embeddings of G are topologically equivalent if they coincide in their outer facial cycle and their set of inner facial cycles. We can thus represent a particular 2-connected embedding up to topological equivalence just by its outer facial cycle and the set of its inner facial cycles. If in addition G is 3-connected, the facial cycles of

$\widetilde{G}$ can be identified combinatorially as the induced non-separating cycles in G. This allows representing any embedding of G just by its outer facial cycle.

Finally, we introduce a notion (as presented in [8]) which plays a fundamental role in graph embedding algorithms following the principle of path addition. Suppose H is a subgraph of G. Then an *H-component* of G is either an edge (together with its ends) in $E(G) \setminus E(H)$ joining two vertices of H or it is a connected component of $G - H$ together with all edges (and their ends) of G joining this component to H. The vertices of an H-component in H are its *vertices of attachment.* If G is 2-connected and C a cycle in G, the vertices of attachment of any C-component of G divide C into edge-disjoint paths, called *segments.* Two C-components *overlap* if not all vertices of attachment of one C-component lie in a single segment of the other C-component. The *overlap graph* of C in G is defined as the graph whose vertices are the C-components of G such that two vertices are adjacent iff the corresponding C-components overlap.

3 Cycle-Constraints

In this section we precisely define cycle-constraints and prove two general results on the problem of finding embeddings satisfying cycle-constraints: a combinatorial characterization and NP-completeness.

Definition 1. *Let G be a graph. A* cycle-constraint *is an expression of the form $x \odot C$, with $x \in G$, cycle $C \subseteq G$ and $x \notin C$. $x \odot C$ is satisfied in a plane graph $\widetilde{G}$ if $\widetilde{x} \in \mathit{int}\ \widetilde{C}$ holds in $\widetilde{G}$. Then, G is* $\mathcal{S}$-embeddable *for a set $\mathcal{S}$ of cycle-constraints if there is an embedding $\widetilde{G}$ of G, called* $\mathcal{S}$-embedding, *which satisfies all the cycle-constraints in $\mathcal{S}$.*

Note that cycle-constraints are topological in the sense that topological transformations of the plane preserve their validity. For the purpose of finding appropriate algorithms it could be useful to have a necessary and sufficient condition for $\mathcal{S}$-embeddability at hand where topological notions are not involved. Besides from what is given in Sect. 2, the following theorem makes use of an easily verifiable observation: if $\widetilde{G}$ is an embedding of a 2-connected planar graph G and C is a cycle of G, then the points of $\widetilde{G}$ contained in Int $\widetilde{C}$ are exactly the vertices of those inner facial cycles of $\widetilde{G}$ whose sum is C.

Theorem 1. *Let G be a graph and $\mathcal{S}$ be a set of cycle-constraints. Then, G is $\mathcal{S}$-embeddable iff there is a 3-connected planar graph G' with $G \subseteq G'$ and $V(G) = V(G')$ and a basis $\mathcal{F}$ of $\mathcal{C}(G')$ consisting of induced non-separating cycles of G' such that for any $x \odot C$ in $\mathcal{S}$: $x \in F$ for some $F \in \mathcal{F}_C$ where $\mathcal{F}_C \subseteq \mathcal{F}$ is defined by $C = \bigoplus_{F \in \mathcal{F}_C} F$.*

Proof. ($\Rightarrow$) Suppose $\widetilde{G}$ is an $\mathcal{S}$-embedding of G. Triangulating $\widetilde{G}$ in the plane leads to an embedding $\widetilde{G'}$ of a maximal planar and thus 3-connected graph G'. The inner facial cycles of $\widetilde{G'}$ are induced non-separating cycles and form a basis

$\mathcal{F}$ of $\mathcal{C}(G')$. Then, if $x \odot C$ holds in $\widetilde{G}$, and thus in $\widetilde{G'}$, x belongs to a cycle of some subset of $\mathcal{F}$ whose sum is C.

($\Leftarrow$) Let G' be the extension of G satisfying the required properties. Then there is an embedding $\widetilde{G'}$ of G' with $\bigoplus_{F\in\mathcal{F}} F$ as the outer facial cycle and all $F \in \mathcal{F}$ as inner facial cycles. According to the observation, $\widetilde{x} \in \operatorname{int} \widetilde{C}$ holds in $\widetilde{G'}$ for all $x \odot C$ in $\mathcal{S}$. Thus, we get an $\mathcal{S}$-embedding $\widetilde{G}$ by restricting $\widetilde{G'}$ to G. □

Applying Theorem 1 requires to determine the coefficients of a given cycle C with respect to a basis $\mathcal{F}$ of $\mathcal{C}(G')$ whose elements are inner facial cycles of some embedding $\widetilde{G'}$ of G'. The following property (see [6]) leads directly to an appropriate procedure working also when G' is 2-connected. If $\mathcal{F}_C$ is the subset of $\mathcal{F}$ whose elements sum up to C, the cycles of $\mathcal{F} \setminus \mathcal{F}_C$ can be ordered as $F_1, \ldots, F_k$ such that C_i is always a cycle and $C_k = C$, where $C_0 := \bigoplus_{F\in\mathcal{F}} F$ and $C_{i+1} := C_i \oplus F_i$. Note that for $F \in \mathcal{F}$ $C_i \oplus F$ is a cycle if and only if $C_i \cap F$ is a non-trivial path in G'. We therefore have to find in each step i a new $F_i \in \mathcal{F}$ so that $C_i \cap F_i$ is a non-trivial path having no edge with C in common. Obviously, this procedure needs polynomial time.

If G is already 3-connected, we can determine whether G is $\mathcal{S}$-embeddable by first finding the set $\mathcal{F}'$ of induced non-separating cycles, e.g. simply by embedding G, and then searching for some $F \in \mathcal{F}'$ satisfying with $\mathcal{F} = \mathcal{F}' \setminus \{F\}$ the condition in Theorem 1. Thus, finding $\mathcal{S}$-embeddings of 3-connected graphs needs polynomial time.

The next theorem proves the general problem of finding $\mathcal{S}$-embeddings to be NP-complete. This holds even for rather simple 2-connected graphs as defined in the proof.

Theorem 2. *The problem of deciding whether a graph G is $\mathcal{S}$-embeddable for some given set $\mathcal{S}$ of cycle-constraints is NP-complete.*

Proof. Membership in NP follows immediately from what is said above: we only have to guess a 3-connected planar supergraph G' of G and verify in polynomial time G' to be $\mathcal{S}$-embeddable.

Next we show the problem to be NP-hard by transformation from the NP-complete problem BETWEENNESS defined as (see [4]): given a finite set A and a set T of ordered triples of distinct elements from A, is there a one-to-one function $f : A \longrightarrow \{1, \ldots, |A|\}$ such that for each $(a, b, c) \in T$ either $f(a) < f(b) < f(c)$ or $f(c) < f(b) < f(a)$ holds? Let $A = \{a_1, \ldots, a_n\}$ and $T = \{t_1, \ldots, t_m\}$ with $t_i = (a_{i_1}, a_{i_2}, a_{i_3})$ be an arbitrary instance of BETWEENNESS. We associate with A the graph G_A defined by $V(G_A) = \{x, y, a_1, \ldots, a_n\}$ and $E(G_A) = \{xa_i, a_iy \mid i \in \{1, \ldots, n\}\}$. For any distinct $i, j \in \{1, \ldots, n\}$ let $C_{i,j}$ denote the cycle x, a_i, y, a_j, x. The set S_T of cycle-constraints encoding T is defined as $S_T := \{a_{i_2} \odot C_{i_1,i_3} \mid t_i = (a_{i_1}, a_{i_2}, a_{i_3}) \in T\}$. Then, obviously, G_A is S_T-embeddable iff there is some function f satisfying the requirements given above. □

4 Restricted Cycle-Constraints

Once a decision problem is proved to be NP-complete, one is left with several possibilities to cope with this problem, one of which is searching for tractable subproblems. In case of graph embeddability with cycle-constraints one can study its complexity with respect to various restrictions on graphs and cycle-constraints. For instance, concerning the connectivity number of graphs we have already determined in Sect. 3 the exact boundary between P and NP: 3-connected graphs make the problem tractable, but a smaller connectivity number leads to NP-completeness. In this section, we restrict graphs to be 2-connected and cycle-constraints to have the property that any two distinct cycles share at most one vertex. In the remainder let G be a 2-connected graph, $\mathcal{S}^0$ be a non-empty set of cycle-constraints of this sort with respect to G and S be the set of cycles occurring in $\mathcal{S}^0$.

We next present in Theorem 3 a necessary and sufficient condition for the existence of $\mathcal{S}^0$-embeddings for 2-connected graphs as well as an $O(n^2)$ algorithm (where $n = |G|$) for finding them. For technical convenience the sufficient condition of this theorem is proved at the end of this section in Proposition 2 where the algorithm is shown to be valid. As it turns out, $\mathcal{S}^0$-embeddability can be characterized with some similarity to the well-known theorem (see [8]) that a graph is planar iff the overlap graph of each cycle is bipartite. For that we first introduce the notion of an $\mathcal{S}^0$-partition.

Definition 2. *Let $C \in S$ and G^C be the overlap graph of C. An $\mathcal{S}^0$-*partition *of G^C is a pair (V_I, V_O) of subsets V_I, V_O of $V(G^C)$ with $V_I \mathbin{\dot\cup} V_O = V(G^C)$ and*

1. *if $XY \in E(G^C)$, then $X \in V_I, Y \in V_O$ or $X \in V_O, Y \in V_I$*
2. *if $x \odot C$ in $\mathcal{S}^0$ and $x \in Y \in G^C$, then $Y \in V_I$.*

Theorem 3. *G is $\mathcal{S}^0$-embeddable iff G is planar and there are some $C_r \in S$ and $\mathcal{S}^0$-partitions (V_I, V_O) of the overlap graph G^C for any cycle $C \in S$ such that for each $C \neq C_r$: $Y \in G^C$ with $C_r \subseteq Y$ is in V_O.*

Proof. ($\Rightarrow$) Let $\widetilde{G}$ be an $\mathcal{S}^0$-embedding of G and select some $C_r \in S$ which is contained in Ext $\widetilde{C}$ of $\widetilde{G}$ for any other cycle $C \in S$. Define for each $C \in S$ an $\mathcal{S}^0$-partition (V_I, V_O) of G^C by $Y \in V_I$ if $Y \subseteq$ Int $\widetilde{C}$ in $\widetilde{G}$, $Y \in V_O$ otherwise.

($\Leftarrow$) (see Proposition 2) □

4.1 Decomposition Trees

The fundamental idea of the embedding algorithm to be developed is to decompose G completely into certain subgraphs that support the control of the $\mathcal{S}^0$-constraints during the embedding process. These subgraphs build the decomposition tree of G. To introduce decomposition trees precisely, we first define the concept of a reduced C-component of a cycle C. The following definition is illustrated in Fig. 1.

Definition 3. *Let B_C be a C-component of G for $C \in S$. If $C' \subseteq B_C$ for some $C' \in S$, let $B_{C,C'}$ be the C'-component of $B_C \cup C$ which contains C. Then,*

$$B_C^r := \begin{cases} B_C \cap \bigcap_{C' \in S, C' \subseteq B_C} (B_{C,C'} \cup C') & \text{if } B_C \text{ contains a cycle of } S \\ B_C & \text{otherwise} \end{cases}$$

is called the reduced C-component *of G in B_C with respect to S.*

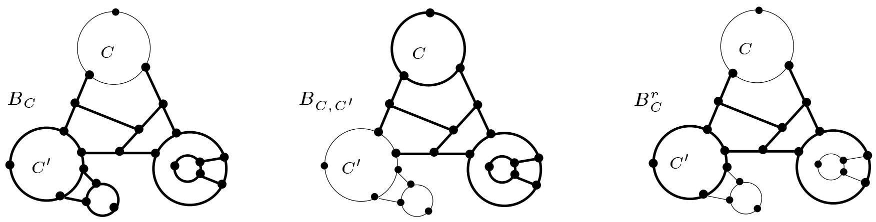

Fig. 1. Construction of the reduced C-component B_C^r from C-component B_C

After removing all C'-components not containing C for cycles $C' \subseteq B_C$, the reduced C-component B_C^r in B_C has the property of having exactly one C'-component in $B_C^r \cup C$ for each $C' \subseteq B_C^r$. Alternatively, we could have defined B_C^r using this property:

Proposition 1. *Let B_C be a C-component of G for $C \in S$ and $\mathcal{B}$ be the set of subgraphs $B \subseteq B_C$ having only one C'-component in $B \cup C$ for all $C' \in S$ with $C' \subseteq B$. Then, B_C^r is exactly the maximal graph in $\mathcal{B}$.*

Note that, by Proposition 1, any C'-component in $B_C^r \cup C$ is already reduced and, more interestingly, any embedding of $B_C^r \cup C$, if any, is completely contained either in Int $\widetilde{C'}$ or Ext $\widetilde{C'}$ for each $C' \subseteq B_C^r \cup C$. This feature of planar reduced components is particularly useful in the embedding procedure.

Corollary 1. *If $B_C^r \cup C$ is planar, then any $C' \in S$ with $C' \subseteq B_C^r \cup C$ is a facial cycle in every embedding of $B_C^r \cup C$.*

If $C' \in S$ is a cycle in B_C^r, each C'-component not including C is removed in B_C^r. However, it contains itself a reduced component as well as the components removed in that one. Thus, the graph G can be recursively decomposed into reduced components with respect to cycles in S. The decomposition tree is a directed tree with root C containing the complete structure of this decomposition process. Fig. 2 (b) and (c) (ignore the labels) show two different decomposition trees of the graph given in (a).

Definition 4. *Let C be a cycle in S. For any $C' \in S$ with $C' \subseteq B_C^r$ let $G_{B_C^r,C'}$ denote the union of C' and all C'-components in $B_C \cup C$ not containing C. If*

S_C^r *is the set of all reduced* C*-components in* G*, the* decomposition tree *of* G *with respect to* C *and* S *is defined as*

$$T_{G,C} := \begin{cases} \{(C,B) \mid B \in S_C^r\} \cup \{(B,C') \mid C' \subseteq B \in S_C^r\} \cup \\ \bigcup_{C' \subseteq B \in S_C^r} T_{G_{B,C'},C'} & \text{if } G \neq C \\ \{G\} & \text{if } G = C. \end{cases}$$

Since the subgraphs $G_{B,C'}$ are again 2-connected, $T_{G,C}$ is well-defined. By definition, all cycles of S are vertices in $T_{G,C}$. Moreover, G is completely decomposed, i.e. $G = \bigcup_{t \in T_{G,C}} t$. This follows inductively from $B_C = B_C^r \cup G_{B_C^r,C_1} \cup \cdots \cup G_{B_C^r,C_k}$ if $C_1, \ldots, C_k$ are all cycles of S in B_C^r.

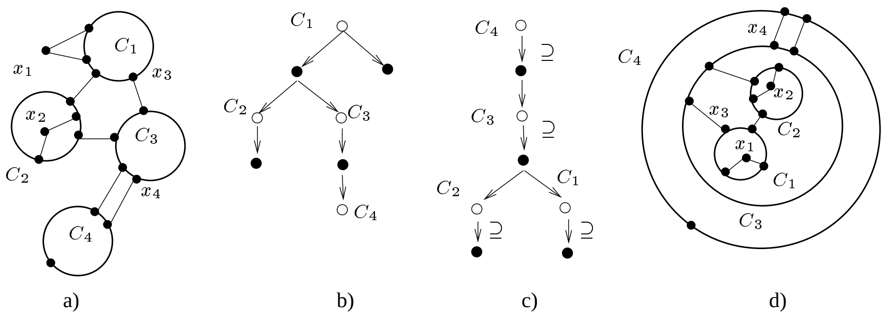

Fig. 2. (a) A graph G with $\mathcal{S}^0 = \{x_i \odot C_i \mid i \in \{1,2,3,4\}\}$; (b) decomposition tree T_{G,C_1}; (c) labelled decomposition tree T^l_{G,C_4}; (d) $\mathcal{S}^0$-embedding of G

The recursive definition of $T_{G,C}$ in Definition 4 leads directly to an algorithm for generating the decomposition tree of G with respect to a cycle C of S. Instead of using Definition 3 to compute the reduced components, one can apply the following informally given procedure. Suppose B_C is a C-component of cycle C and A the set of its vertices of attachment. Start a depth-first search of B_C beginning at some vertex of A. Whenever a cycle $C' \in S$, $C' \neq C$, appears for the first time, determine all C'-components in B_C not including all vertices of A and remove them from B_C. The union of these C'-components together with C' is actually $G_{B_C^r,C'}$. When the search is finished, the subgraph denoted by B_C is now the reduced C-component. Using Proposition 1 one can easily verify that the whole procedure computes $T_{G,C}$ correctly. Moreover, if G is planar, it only needs $O(n^2)$ time where $n = |G|$ since the number of edge-disjoint cycles in G cannot exceed n.

4.2 Embedding Algorithm

Provided with Theorem 3 and the concept of decomposition trees, we are now able to describe an $O(n^2)$ algorithm for constructing an $\mathcal{S}^0$-embedding of G if

G has any. Instead of a particular drawing the algorithm generates a formal representation of an $\mathcal{S}^0$-embedding consisting of its outer facial cycle and the set of its inner facial cycles.

Step 1. First, we must check G to satisfy the necessary condition of Theorem 3 for being $\mathcal{S}^0$-embeddable and thereby determine the particular cycle $C_r \in S$. That means we test planarity and determine the overlap graph G^C of any cycle $C \in S$. For each $C \in S$ we consider the components H of G^C: find an $\mathcal{S}^0$-partition (V_I, V_O) of H, if any, and define $M_H = V_O$ if $x \in Y \in H$ for some constraint $x \odot C$, otherwise $M_H = V_I \cup V_O$. Let M_C be the union of all sets M_H for components H of G^C; M_C contains the C-components that can be embedded outside from C and therefore are candidates for the choice of C_r. Finally, we have to find a $C_r \in S$ which is in M_C of any other cycle $C \in S$. The cycle C_r satisfies the condition of Theorem 3 and is used furthermore to construct an $\mathcal{S}^0$-embedding of G. If there is no such cycle or some H has no $\mathcal{S}^0$-partition, we can stop trying to embed G. Since the number of cycles in S cannot exceed n, this procedure needs $O(n^2)$ time.

Step 2. Having found the cycle C_r, we compute the decomposition tree T_{G,C_r} with root C_r as described in Sect. 4.1. Next we encode the $\mathcal{S}^0$-constraints directly into T_{G,C_r} by labelling its edges. Suppose $x \odot C$ is a constraint in $\mathcal{S}^0$. Then, by definition of C_r, there must be a path $C, B, \ldots, D$ from C to some D in T_{G,C_r} with $x \in D$. We thus label the edge (C, B) with $\supseteq$ since in any $\mathcal{S}^0$-embedding of G the reduced C-component B must be in the interior of C. When all constraints in $\mathcal{S}^0$ are encoded in this way, the labelled decomposition tree is denoted by T^l_{G,C_r}; an example is shown in Fig. 2 (c).

Step 3. In the last step we obtain with algorithm *embedding* given in Fig. 3 a representation of an $\mathcal{S}^0$-embedding. Roughly speaking, this algorithm is of the same spirit as the path addition procedure originally presented by Demoucron et al. [2] and also briefly described in [1]. As G and C_r satisfy the necessary condition of Theorem 3, algorithm *embedding* generates a set $\mathcal{F}$ of cycles of G and a set $\mathcal{F}^I$ of sets $\mathcal{F}^I_C \subseteq \mathcal{F}$ for each $C \in S$ such that for any $F \in \mathcal{F}$ contained in no $\mathcal{F}^I_C$ there is some $\mathcal{S}^0$-embedding of G with F as the outer facial cycle and $\mathcal{F} \setminus \{F\}$ as the set of inner facial cycles.

We now explain the algorithm in more detail. The parameters C, G', $\mathcal{F}$ and $\mathcal{F}^I$ have the following meaning: C is a cycle in S and the currently visited vertex of the depth-first search through T^l_{G,C_r}; G' is the subgraph of G processed so far ($C \subseteq G'$); $\mathcal{F}$ is a set of facial cycles of some embedding of G' and $\mathcal{F}^I$ a set of sets $\mathcal{F}^I_{C'} \subseteq \mathcal{F}$ for each visited cycle $C' \in T^l_{G,C_r}$. The embedding procedure must be started with the call *embedding*$(T^l_{G,C_r}, C_r, C_r, \{C_r\}, \{\mathcal{F}^I_{C_r}\})$ where $\mathcal{F}^I_{C_r} = \{C_r\}$. Suppose C, G', $\mathcal{F}$ and $\mathcal{F}^I$ are the current variables. Algorithm *embedding* expand $\mathcal{F}$ and $\mathcal{F}^I$ with all reduced C-components B. In choosing which B to take first, those B are preferred that can be attached to only one facial cycle

Algorithm *embedding* $(T^l_{G,C_r}, C, G', \mathcal{F}, \mathcal{F}^I)$

Determine the set $\mathcal{M}$ of B_i where (C, B_i) is an edge in T^l_{G,C_r};
while $\mathcal{M} \neq \emptyset$ **do**
    for all $B_i \in \mathcal{M}$ **do**
        Determine the set $\mathcal{F}_{B_i}$ of cycles from $\mathcal{F}$
        containing all vertices of $C \cap B_i$;
        if (C, B_i) has label $\supseteq$ **then** $\mathcal{F}_{B_i} \leftarrow \mathcal{F}_{B_i} \cap \mathcal{F}^I_C$;
    if $|\mathcal{F}_{B_i}| = 1$ for some $B_i \in \mathcal{M}$
        then $B \leftarrow B_i$ and $F \in \mathcal{F}_{B_i}$
        else $B \leftarrow B_i$ for some B_i and $F \in \mathcal{F}_{B_i}$;
    $G' \leftarrow G' \cup B$;
    Determine the set $\mathcal{F}'$ of facial cycles of an embedding of $F \cup B$;
    if $\mathcal{F} = \{C_r\}$ **then** $\mathcal{F} \leftarrow \mathcal{F}'$
        else $\mathcal{F} \leftarrow (\mathcal{F} \cup \mathcal{F}') \setminus \{F\}$;
    for all $\mathcal{F}^I_{C'} \in \mathcal{F}^I$ **do**
        if $F \in \mathcal{F}^I_{C'}$ **then** $\mathcal{F}^I_{C'} \leftarrow (\mathcal{F}^I_{C'} \cup \mathcal{F}') \setminus \{F\}$;
    for all C' with (B, C') in T^l_{G,C_r} **do**
        $\mathcal{F}^I_{C'} \leftarrow \{C'\}$;
        $\mathcal{F}^I \leftarrow \mathcal{F}^I \cup \{\mathcal{F}^I_{C'}\}$;
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \{B\}$;
for all C' with $\{C'\} \in \mathcal{F}^I$ **do**
    embedding$(T^l_{G,C_r}, C', G', \mathcal{F}, \mathcal{F}^I)$;
return G', $\mathcal{F}$, $\mathcal{F}^I$;

Fig. 3. Algorithm *embedding* for generating facial cycles of an $\mathcal{S}^0$-embedding

of $\mathcal{F}$. Otherwise an arbitrary reduced C-component B and one of its two facial cycles F it can be attached to are chosen. If the edge from C to B in T^l_{G,C_r} is labelled, B must be attached to a cycle in $\mathcal{F}^I_C$. Then the facial cycles of an embedding of $B \cup F$ are determined (note that G was checked to be planar). By Corollary 1, all cycles of S contained in $B \cup F$ are facial cycles. Thus we must update $\mathcal{F}$ and all $\mathcal{F}^I_{C'}$ in $\mathcal{F}^I$ by replacing F with all new facial cycles of $B \cup F$ except F itself. Moreover, $\mathcal{F}^I$ must be extended by the new sets $\mathcal{F}^I_{C'}$ for all $C' \in S$ with $C' \subseteq B$. After attaching all B, the algorithm is applied in a depth-first manner to all cycles C' which are successors of B in T^l_{G,C_r}.

To establish the validity of this algorithm in the next proposition, let for the ith procedure call, $i = 1, \ldots, k := |S|$, C_i denote the variable C and $G'_i, \mathcal{F}_i, \mathcal{F}^I_i$ denote the variables $G', \mathcal{F}, \mathcal{F}^I$, respectively, at the time when the while-loop ends.

Proposition 2. *Suppose G is planar and the overlap graph G^C of any cycle $C \in S$ has an $\mathcal{S}^0$-partition (V_I, V_O) such that for $C \neq C_r$: $Y \in G^C$ with $C_r \subseteq Y$ is in V_O. Then, algorithm* embedding *terminates with $G'_k = G$, $\mathcal{F}_k$ and $\mathcal{F}^I_k$ such that for any $F_0 \in \mathcal{F}_k$ contained in no $\mathcal{F}^I_C$ of $\mathcal{F}^I_k$ there is some $\mathcal{S}^0$-embedding of G with F_0 as the outer facial cycle and $\mathcal{F}_k \setminus \{F_0\}$ as the set of inner facial cycles.*

Proof. In the sequel we abbreviate the union of all cycles in $\mathcal{F}_C^I \in \mathcal{F}_i$ by H_C^i. Then the proposition can be proved by showing for any $i \in \{1, \ldots, k\}$:

1. $\mathcal{F}_i$ is the set of facial cycles of some embedding of G'_i
2. $C = \bigoplus_{F \in \mathcal{F}_C^I} F$ for any $\mathcal{F}_C^I \in \mathcal{F}_i^I$
3. $B \subseteq H_{C_j}^i$, $1 \leq j \leq i$, for each successor B of C_j with labelled edge (C_j, B)
4. $C_r \not\subseteq H_C^i$ for any $C \in S$, $C \neq C_r$, with $C \subseteq G'_i$.

Let $i = 1$, i.e. $C_1 = C_r$. Since G^{C_r} has an $\mathcal{S}^0$-partition (V_I, V_O), the algorithm generates $G'_1, \mathcal{F}_1, \mathcal{F}_1^I$ by stepwise adding the successors B of C_r such that for some $\mathcal{S}^0$-partition (V'_I, V'_O) of G^{C_r} (which may be different from (V_I, V_O)): $B \subseteq Y \in V'_I$ iff $B \subseteq H_{C_r}^1$. Thus 3) is satisfied. By Corollary 1, for each B and F chosen in the algorithm any cycle $C \in S$ in $B \cup F$ belongs to the set $\mathcal{F}'$ of facial cycles of some embedding of $B \cup F$ found by the algorithm. This and the way how $\mathcal{F}_1$ and $\mathcal{F}_1^I$ are updated ensure 1), 2) and 4) to be valid.

Suppose $i > 1$. By assumption, G^{C_i} has an $\mathcal{S}^0$-partition (V_I, V_O) with $Y \in V_O$ if $C_r \subseteq Y$. Since 4) holds for $i-1$, the algorithm extends $G'_{i-1}, \mathcal{F}_{i-1}, \mathcal{F}_{i-1}^I$ to $G'_i, \mathcal{F}_i, \mathcal{F}_i^I$ by adding the successors B of C_i such that for some $\mathcal{S}^0$-partition (V'_I, V'_O) of G^{C_i} with $Y \in V'_O$ if $C_r \subseteq Y$: $B \subseteq Y \in V'_I$ iff $B \subseteq H_{C_i}^i$. This shows 3); and 1), 2) and 4) can be proved analogously to what is said above. $\square$

We finally note that this procedure needs only $O(n^2)$ time because the number of calls is $|S| \leq n$ and each while-loop needs $O(n)$ time.

Acknowledgments

The author would like to thank Bernhard Nebel and Jochen Renz for their helpful comments.

References

1. J.A. Bondy, U.S.R. Murty: *Graph Theory with Applications*, MacMillan, 1976
2. G. Demoucron, Y. Malgrange, R. Pertuiset: Graphes planaires: reconnaissance et construction de représentations planaires topologiques, *Revue Française de Recherche Opérationelle*, 8: 33-47, 1964
3. R. Diestel: *Graph Theory*, Springer, 1997
4. M.R. Garey, D.S. Johnson: *Computers and Intractability – A Guide to the Theory of NP-Completeness*, Freeman, 1979
5. W. He, K. Marriott: Constrained graph layout, in S. North (ed.), *Graph Drawing*, Proceedings of GD'96, LNCS 1190, 217-232, 1997
6. M.M. Syslo: An efficient cycle vector space algorithm for listing all cycles of a planar graph, *SIAM Journal on Computing*, 10,4: 797-808, 1981
7. R. Tamassia: Constraints in graph drawing algorithms, *Constraints*, 3: 87-120, 1998
8. C. Thomassen: Planarity and duality of finite and infinite graphs, *Journal of Combinatorial Theory, Series B*, 29: 244-271, 1980