Planar Graphs, via Well-Orderly Maps and Trees

Nicolas Bonichon*, Cyril Gavoille*, Nicolas Hanusse*,
Dominique Poulalhon†, and Gilles Schaeffer‡

* Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Université Bordeaux 1, 33405 Talence Cedex, France
{bonichon, gavoille, hanusse}@labri.fr

† Laboratoire d'Informatique Algorithmique, Fondements et Applications (LIAFA)
case 7014, 2, place Jussieu, 75251 Paris Cedex 05, France
dominique.poulalhon@liafa.jussieu.fr

‡ Laboratoire d'Informatique de l'École Polytechnique (LIX)
École polytechnique, 91128 Palaiseau Cedex, France
gilles.schaeffer@lix.polytechnique.fr

Abstract. The family of well-orderly maps is a family of planar maps with the property that every connected planar graph has at least one plane embedding which is a well-orderly map. We show that the number of well-orderly maps with n nodes is at most $2^{\alpha n+O(\log n)}$, where $\alpha \approx 4.91$. A direct consequence of this is a new upper bound on the number $p(n)$ of unlabeled planar graphs with n nodes, $\log_2 p(n) \leqslant 4.91n$.
The result is then used to show that asymptotically almost all (labeled or unlabeled), (connected or not) planar graphs with n nodes have between $1.85n$ and $2.44n$ edges.
Finally we obtain as an outcome of our combinatorial analysis an explicit linear time encoding algorithm for unlabeled planar graphs using, in the worst-case, a rate of 4.91 bits per node and of 2.82 bits per edge.

Keywords: planar graph, triangulation, realizer, well-orderly map, coding

1 Introduction

Counting the number of (non-isomorphic) planar graphs with n nodes is a well-known long-standing unsolved graph-enumeration problem (cf. [LW87]). There is no known closed formula, neither asymptotic for unlabeled planar graphs.

There are only upper and lower bounds on the *growth rate* of the sequence of numbers $p(n)$ of unlabeled planar graphs. This growth rate, defined as $\mu = \lim_{n\to\infty} p(n)^{1/n}$, currently ranges between 27.2268 and 32.1556 (a superadditivity argument shows that such a limit exists [DVW96,MSW01]).

The lower bound on μ comes from an asymptotic on the number of labeled planar graphs. This asymptotic is on the form $n!\lambda^{n+o(n)}$ [DVW96,MSW01], and in [GN04], a precise estimation of λ is given: $27.2268 < \lambda < 27.2269$. The upper bound on μ, due to [BGH03], comes from succinct encoding of plane planar graphs. More precisely, after a suitable embedding and triangulation of the planar graph, it is shown that such embeddings can be represented by a binary string of length at most $5.007n$ bits. Such representation implies that $p(n) \leqslant 2^{5.007n} \approx (32.1556)^n$.

Technically, enumerating unlabeled graphs is more difficult than counting the labeled version. And, as pointed out in [BGW02], almost all labeled 2- and 1-connected planar graphs have exponentially large automorphism groups. In other words, Wright's Theorem [Wri71] does not hold for random planar graphs, the asymptotic number of labeled and unlabeled planar graphs differ in more than the $n!$ term, i.e., $\lambda < \mu$. So, an asymptotic on the number of labeled planar graphs would not give a sharp lower bound on the growth rate of $p(n)$. The situation from the upper bound side is not better. There are many ways to embed a planar graph, and to recover the graph from a suitable triangulation requires deep understanding of plane triangulations, in particular their enumeration given several parameters depending on the input graph.

Besides the pure combinatorial aspect, the “encoding” approach is also relevant in Computer Science where a lot of attention is given to efficiently represent discrete objects. At least two field of applications of high interests are concerned with succinct planar graph representation: Computer Graphics [KADS02,KR99,Ros99] and Networking [FJ89,GH99,Lu02,Tho01].

1.1 Related works

Obviously, without sharp asymptotic formula, properties and behavior of large random objects cannot be described precisely. For lack of an adequate model, very little is known on random planar graphs. However, random generation of planar graphs has been investigated in the last decade.

Using a simple Markov chain, Denis et al. [DVW96] showed, that, experimentally, random labeled planar graphs have $2n$ edges. In fact, Bodirsky et al. [BGK03] have designed the first polynomial time (uniform) random generator of labeled planar graphs. Although limited in their experiments (mainly by the time complexity of this algorithm), they showed that actually the number of edges in a random labeled planar graph is more than $2n$. The best proved bounds on the number of edges in a random labeled planar graph are $1.85n$ [GM02] and $2.54n$ [BGH03], for the unlabeled case $1.70n$ and $2.54n$, by [BGH03].

Succinct representation of n-node m-edge planar graphs has a long history. Turán [Tur84] pioneered a $4m$ bit encoding, that has been improved later by Keeler and Westbrook [KW95] to $3.58m$. Munro and Raman [MR97] then proposed a $2m + 8n$ bit encoding based on the 4-page embedding of planar graphs (see [Yan89]). In a series of articles, Lu et al. [CLL01,CGH$^+$98] refined the coding to $4m/3 + 5n$ thanks to orderly spanning trees, a generalization of Schnyder's trees [Sch90].

1.2 Our results

Any planar embedding of a n-node planar graph with n nodes can be seen as a subgraph of a n-node triangulation of the plane. Once given a triangulation and a set of edges to keep (or to remove), a planar map and the corresponding graph can be constructed. The converse is false in general. There is no known method to uniquely associate a triangulation to a planar graph.

However, in [BGH03], a linear time algorithm is given to construct a triangulation of the plane in a canonical way for any planar graph, once given a planar embedding. The reader should keep in mind that there is a-priori no unique embedding for a planar graph. Some of planar embeddings have interesting graph properties based on the Schnyder's partition [Sch90] of triangulations into trees. A new class of planar embeddings are proposed in [BGH03]: the *well-orderly maps*, a more restrictive version of the *orderly maps* of Chuang et al. [CLL01]. The two main properties of well-orderly maps that can be exploited for graph coding are: 1) every planar graph admits such an embedding, and 2) given a well-orderly map, we can uniquely associate a triangulation.

The main result of this paper is to give a good approximation of the number of well-orderly maps. As a byproduct, it gives a new upper bound on the number of planar graphs. More interestingly, the combinatorial analysis allow to us to give an explicit coding of such maps (and thus of planar graphs), as a function of n and m, the number of edges: 4.91 bits per node, and 2.82 bit per edge (clearly, $2.82m$ bits is always smaller than $4m/3 + 5n$ bits). It follows also a new bound on the number of edges of a random planar graph (labeled or not).

The paper is organized as follows. We describe in Section 2 the relationships between well-orderly maps, super-triangulations and realizers. The new coding is presented in Section 3, and in Section 4 are given the applications to the number of unlabeled planar graphs and to the number of edges in random planar graphs. Another application of our results is an upper bound on the minimal grid area of random triangulation of the plane. We show that in average, plane triangulations can be drawn on grids of area at most $\frac{7}{8}n \times \frac{7}{8}n$ and $\frac{11}{16}n \times \frac{5}{6}n$.

2 Encoding planar graphs with minimal realizers

In this section we collect some results from [BGH03] about planar graphs, well-orderly maps, super-triangulations and realizers. In the last paragraph, these results are used to prove a new representation theorem.

2.1 Planar graphs and well-orderly maps

A *planar map* (or *plane graph*) is an embedding of a connected planar graph on the plane so that edges meet only at their endpoints. When cutting the plane along the edges, the remaining components are called the faces. Apart from the unbounded component, all these faces are homeomorphic to discs. A planar map is *rooted* if one of its edges is distinguished and oriented. This determines a *root edge*, a *root node* (its origin) and a *root face* (to its left), also called *external face* or *outerface*. A *triangulation* of the plane (or a maximal plane graph) is a planar map without loops or multiple edges such that all faces are triangles.

A plane tree is, as usual, a rooted tree such that the siblings of a node are linearly ordered. Equivalently it is a planar map with one face. Among the nodes of a tree, we distinguish the root, the inner nodes and the leaves. A spanning tree of a planar map is a subset of edges that forms a tree connecting all nodes.

Let T be a rooted spanning tree of a planar map H, and let $v_1, \ldots, v_n$ be the clockwise preordering of the nodes in T. Two nodes are *unrelated* if neither of them is an ancestor of the other in T. An edge of H is unrelated if its endpoints are unrelated.

A node v_i is *orderly* in H with respect to T if the edges incident to v_i in H form the following four (possibly empty) blocks in clockwise order around v_i (see Fig. 1(b)):

- $B_P(v_i)$: the edge incident to the parent of v_i in T;
- $B_<(v_i)$: edges that are unrelated in T and incident to nodes v_j with $j < i$;
- $B_C(v_i)$: edges that are incident to the children of v_i in T; and
- $B_>(v_i)$: edges that are unrelated in T and incident to nodes v_j with $j > i$.

A node v_i is *well-orderly* if it is orderly and if the clockwise first edge $(v_i, v_j) \in B_>(v_i)$, if it exists, verifies that the parent of v_j is an ancestor of v_i.

A rooted spanning tree T of H is a *well-orderly tree* of H if all the nodes of T are well-orderly in H with respect to T. A planar map H is a *well-orderly map of root* v if it contains a well-orderly tree of root v.

Theorem 1 ([BGH03]). *Let G be a connected planar graph, and let v be any node of G. Then G has a well-orderly map of root v, which can be computed in linear time. Moreover, a well-orderly map of root v has a unique well-orderly tree of root v, which can also be computed in linear time.*

In Fig. 2 two orderly trees $\overline{T}_0$ span the same triangulation but only one is the well-orderly tree. Observe that by definition of well-orderly nodes, an edge of H which is related with respect to a well-orderly tree T (i.e. one endpoint is a descendant of the other one in T) must belong to the tree T: indeed all edges are either unrelated or connect a node to its father. In particular all the edges incident in H to the root of T are in T.

2.2 Minimal realizers and super-triangulations

A *realizer* of a triangulation is a partition of its interior edges (the edges that do not lie on the external face) into three sets T_0, T_1, T_2 of directed edges such that the following conditions hold for each interior node v (see Fig. 1(a)):

- the clockwise order of the edges incident with v is: leaving in T_0, entering in T_1, leaving in T_2, entering in T_0, leaving in T_1 and entering in T_2;
- there is exactly one leaving edge incident with v in each of the sets T_0, T_1, and T_2.

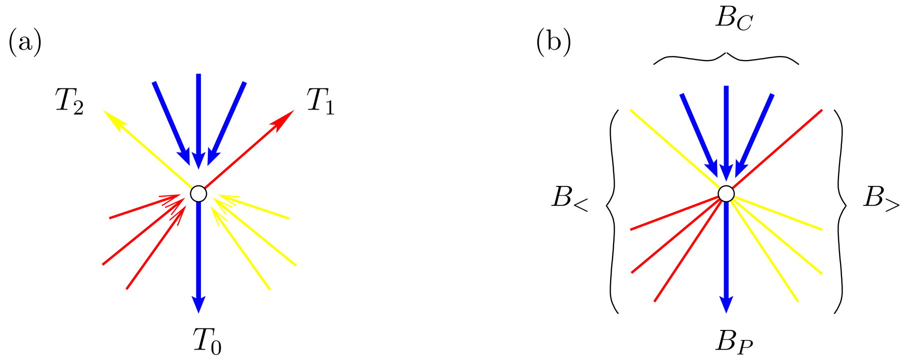

Fig. 1. Relationship between realizer and orderly tree: (a) edge-orientation rule around a node for a realizer, and (b) blocks ordering around an orderly node.

Hereafter, when $R = (T_0, T_1, T_2)$ is a realizer, R also denotes the underlying triangulation.

Observe that if (T_0, T_1, T_2) is a realizer, then (T_1, T_2, T_0) and (T_2, T_0, T_1) are also realizers. This cyclic permutation of the three sets of edges does not in general provide all the distinct realizers of a given triangulation. Fig. 2 depicts two realizers for a same triangulation.

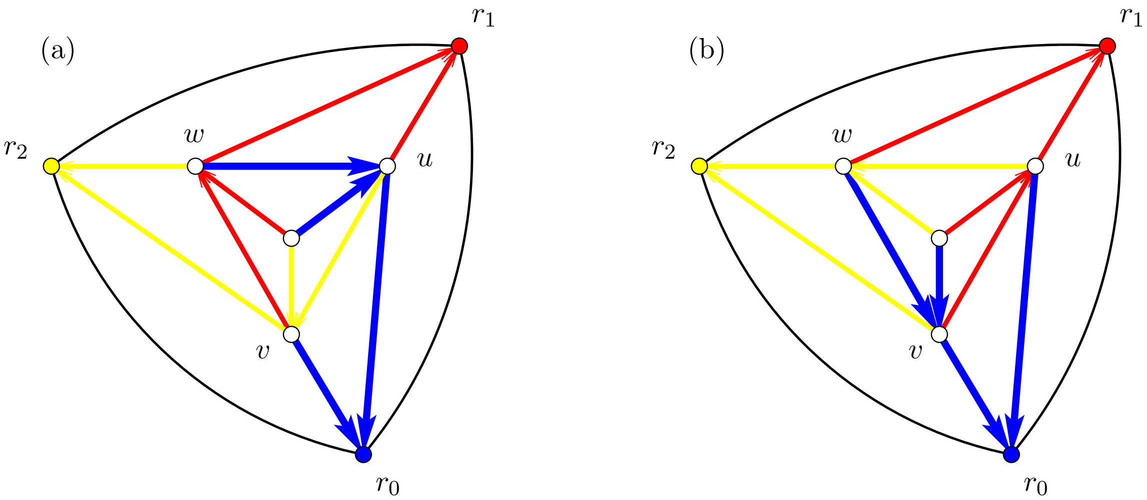

Fig. 2. Two realizers for a triangulation. The tree $\overline{T}_0$ rooted in r_0 (the tree with bold edges augmented with the edges (r_0, r_1) and (r_0, r_2)) is well-orderly in (b), but only orderly in (a) since node v is not well-orderly.

Schnyder showed in [Sch90] that if (T_1, T_2, T_3) is a realizer then each set T_i induces a tree rooted in one node of the external face and spanning all interior nodes. Moreover, for each T_i, we denote by $\overline{T}_i$ the tree composed of T_i augmented with the two edges of the external face incident to the root of T_i. For every non-root node $u \in T_i$, we denote by $p_i(u)$ the parent of u in T_i.

A realizer $S = (T_0, T_1, T_2)$ is a *super-triangulation* of a graph G if:

1. $V(S) = V(G)$ and $E(G) \subseteq E(S)$;
2. $E(T_0) \subseteq E(G)$;
3. $\overline{T}_0$ is a well-orderly tree of S; and
4. for every inner node v of T_2, $(v, p_1(v)) \in E(G)$.

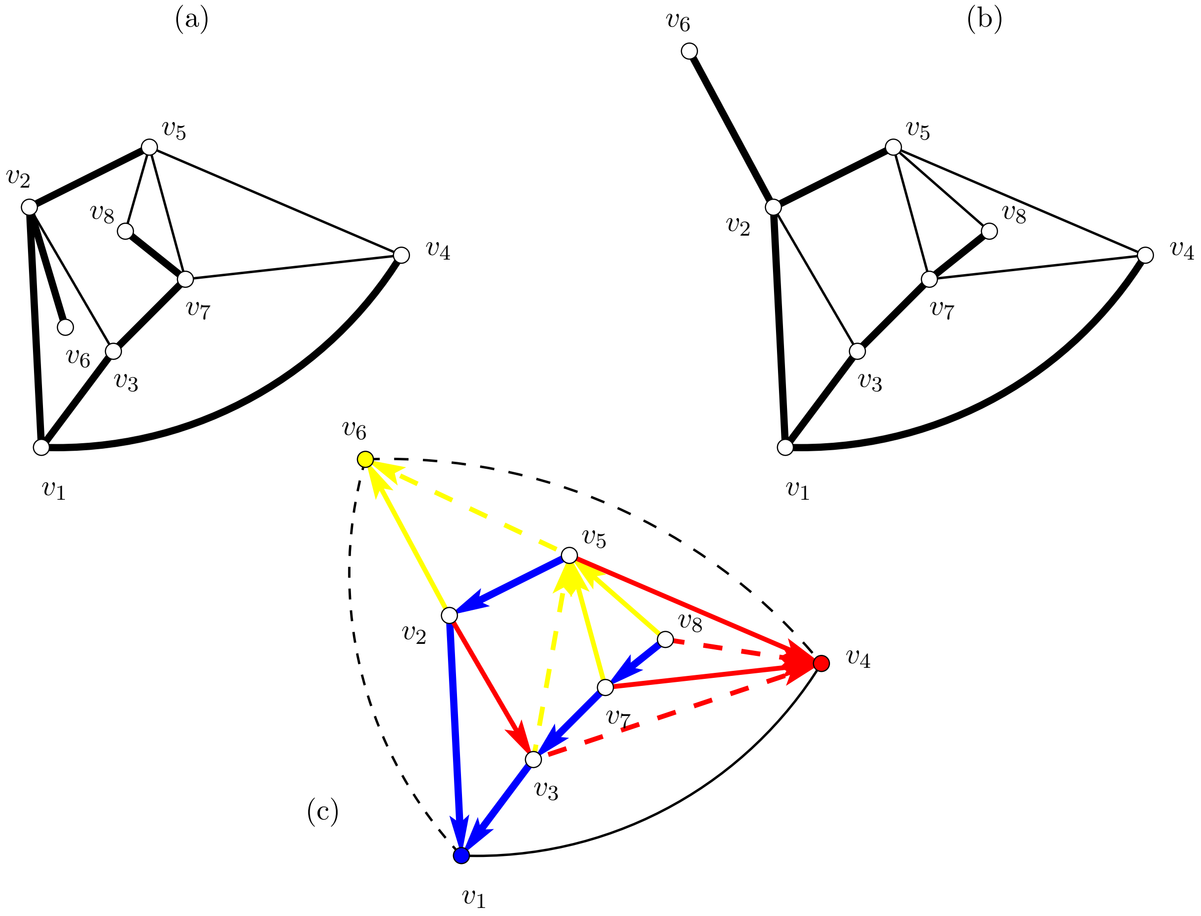

Fig. 3. (a) A planar graph G with an embedding which is not well-orderly. An easy way to see that it is not a well-orderly, is to remark that the edges $(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_6)$ must be in any spanning tree of G rooted at v_1 such that G has only parent edges and unrelated edges. In such trees, v_2 is clearly not an orderly node. (b) A well-orderly map of G. (c) A super-triangulation of G (dotted edges are not in G).

Lemma 1 ([BGH03]). *Let H be a well-orderly map, and T its unique well-orderly tree of root r_0. Assume that T has at least two leaves. Let r_2 and r_1 be the clockwise first and last leaves of T respectively. Then, there is a unique super-triangulation (T_0, T_1, T_2) of the underlying graph of H, preserving the embedding H, and such that each T_i has root r_i. Moreover, $T_0 = T \setminus \{r_1, r_2\}$ and the super-triangulation is computable in linear time.*

There is an alternative characterization of super-triangulation in terms of minimal realizers. A *cw-triangle* (or clockwise triangle), is a triple of nodes (u, v, w) (not necessarily corresponding to a face) of a realizer such that $p_2(u) = v$, $p_1(v) = w$, and $p_0(w) = u$. A *minimal realizer* is a realizer that does not contain any clockwise triangle. In the realizer depicted in Fig. 2(a), (u, v, w) forms a cw-triangle, whereas the realizer of Fig. 2(b) has no cw-triangle.

Lemma 2 ([BGH03]). *Let $S = (T_0, T_1, T_2)$ be any realizer. The following statements are equivalent:*

1. *S is a super-triangulation for some graph G.*
2. *S is a minimal realizer.*
3. *The tree $\overline{T}_i$ is well-orderly in S, for every $i \in \{0, 1, 2\}$.*

2.3 Results of the paper

Theorem 2 (Coding version). *The following encoding sequence hold:*

- *Any connected planar graph can be embedded as a well-orderly map.*
- *Any well-orderly map can be represented as a minimal realizer (T_1, T_2, T_3) with a subset of marked edges included in the sets of edges of T_2 and of edges (u, v) of T_1 such that u is a leaf of T_1.*

Our first new result in this paper is that in fact the second encoding is almost tight.

Theorem 3 (Counting version). *Let H_n (resp. $H_{n,m}$) denote the set of well-orderly maps with n nodes (resp. with n nodes and m edges), and $R_{n,\ell}$ denote the set of minimal realizers (T_0, T_1, T_2) with n nodes and l leaves in T_2. Then*

$$\frac{1}{8}\sum_{\ell=1}^{n-3} |R_{n,\ell}| 2^{n+\ell} \leqslant |H_n| \leqslant \sum_{\ell=1}^{n-3} |R_{n,\ell}| 2^{n+\ell}. \tag{1}$$

$$\frac{1}{8}\sum_{\ell=\max\{1,2n-m-6\}}^{n-3} |R_{n,\ell}| \binom{n+\ell}{m-2n+6+\ell} \leqslant |H_{n,m}| \leqslant \sum_{\ell=\max\{1,2n-m-6\}}^{n-3} |R_{n,\ell}| \binom{n+\ell}{m-2n+6+\ell}. \tag{2}$$

Proof (Theorem 3). Let $S = (T_0, T_1, T_2)$ be an element of $R_{n,\ell}$, and G be a connected planar graph such that S is a super-triangulation of G i.e. $E(T_0) \subseteq E(G)$. The number of edges of a triangulation of n nodes is $3n-6$. Among the $3n-6$ edges of S, there are $(n-3)$ edges that belong to T_0 and $n-3-\ell$ edges $(v, p_1(v))$ such that v is an inner node of T_2 (remind that T_i does not contain the roots of $T_{j\neq i}$). All these edges belong also to G (see super-triangulations). In S there are $n+\ell$ other edges, so there are at most $2^{n+\ell}$ subgraphs of S satisfying the previous conditions and $\binom{n+\ell}{m-2n+6+\ell}$ m-edge subgraphs of S also satisfying the previous conditions. This leads to the upper bounds.

As a well-orderly map admits a unique super-triangulation (see Lemma 1), the lower bounds in Theorem 3 follow if we prove that for all realizer $S \in R_{n,j}$, the number of well-orderly maps that admit S as a super-triangulation is at least $2^{n+\ell-3}$, among which $\binom{n+\ell}{m-2n+6+\ell}$ have m edges. Let

$$E' \;=\; E(S) \setminus \big(E(\overline{T}_0) \cup \{(v, p_1(v)) \mid v \text{ is an inner node of } T_2\} \cup \{(r_1, r_2)\}\big)$$

Since the cardinality of E' is $(3n-6)-(n-1)-(n-3-\ell)-1 = n+\ell-3$, it is sufficient to prove that removing any subset of edges of E' we obtain a different well-orderly map. First observe that removing different subsets of edges, we clearly obtain different maps since the spanning tree T_0 is always kept. It remains to check the well-orderly condition.

Since S is a well-orderly map, the property is true when no edges are removed. Let us assume that the submap G_1 of S obtained by removing some edges of E' is well-orderly and consider the submap G_2 obtained by removing one more edge $(u, v) \in E'$. In G_1, $\overline{T}_0$ is a well orderly tree, and (u, v) is unrelated edge with respect to $\overline{T}_0$, so that $\overline{T}_0$ is an orderly spanning tree of G_2. It remains to check that u and v are well-orderly. We distinguish two cases:

- $(u, v) \in T_2$: node v was an inner node of the tree T_2 in G_1, hence the edge $e' = (v, p_1(v))$ belongs to G_1 and to G_2. Since edge e' is the clockwise first edge of $B_>(v)$ and node $p_0(p_1(v))$ is still an ancestor of v in T_0, so v is well-orderly. As for node u, since no edge of block $B_>(u)$ has changed between G_1 and G_2, u is still well-orderly.
- $(u, v) \in T_1$: this implies that u is a leaf of the tree T_2 in G_1 and in G_2. It follows that $B_>(u) = \{(u, v)\}$ in G_1 and $B_>(u) = \varnothing$ in G_2. By definition, u (and so $\overline{T}_0$ since $B_>(v)$ is the same in G_1 and in G_2) is well-orderly.

□

3 Counting and coding trees

In this section we briefly recall a result from [PS03] about minimal realizers and plane trees. An encoding of well-orderly maps follows.

3.1 Minimal realizers and plane trees

A tree is planted if it is rooted on a leaf. Let $\mathcal{B}_n$ be the set of planted plane trees with n nodes and $2n$ leaves such that each node is adjacent to 2 leaves. Given a planted tree T of $\mathcal{B}_n$, its canonical orientation shall be toward the root for all inner edges, and toward the leaf for all dangling edges.

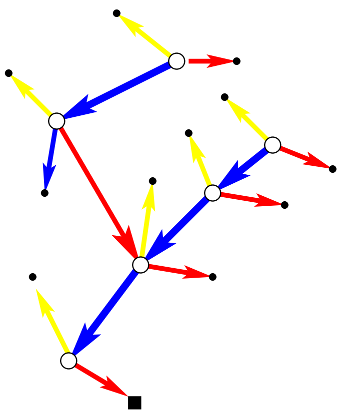

Fig. 4. A planted tree of $\mathcal{B}_n$ (the root is indicated by a square).

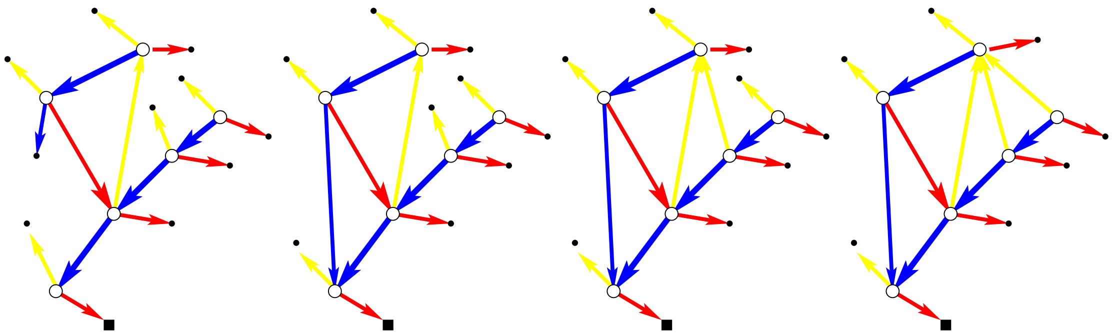

Fig. 5. The partial closure of the tree of Fig. 4.

A triple (e_1, e_2, e_3) of edges of a map M is an *admissible triple* if $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$ and $e_3 = (v_2, v_3)$ appear consecutively in the clockwise direction around the infinite face and if v_3 is a leaf. The *local closure* of M at the admissible triple (e_1, e_2, e_3) is obtained by fusing the leaf v_3 on node v_0 so as to create triangular face. Observe that by construction the orientation of the dangling edge prevents the formation of cw-triangles.

The *local closure* of a tree T of $\mathcal{B}_n$ is the map obtained by performing iteratively the local closure of any available admissible triple in a greedy way. As shown in [PS03], the local closure is well defined independently of the order of local closures. Moreover all bounded faces of the

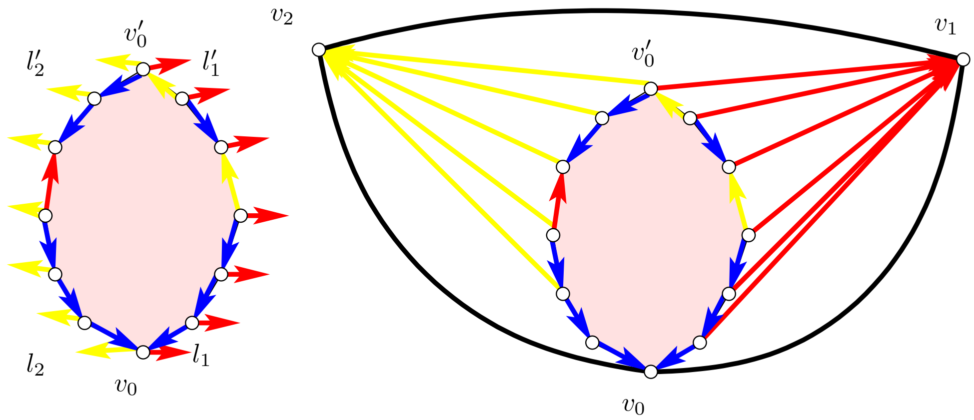

Fig. 6. The structure after a partial closure, and the complete closure.

resulting map are triangular and the outer face has the structure shown on Fig. 6 (left hand side). In particular there are exactly two *canonical* dangling edges in the infinite face that are immediately followed by dangling edges in the clockwise direction around the infinite face. A tree T is *balanced* if its root is one of the two canonical leaves. Finally, the *complete closure* of a balanced tree T is the map obtained from the partial closure of T by fusing each remaining non canonical leaf with following canonical leaf in clockwise direction and adding a root edge, as illustrated by Fig. 6 (right hand side).

Theorem 4 ([PS03]). *Complete closure is one-to-one correspondence between balanced trees with $n-2$ and triangulations with n nodes. Moreover, the orientation of inner edges of the triangulation that is induced by the tree corresponds, via the coloration rule of Fig. 1(a) to a minimal realizer of the triangulation.*

Observe that the color of edges can be deduced from their orientation directly on the balanced tree from the application of the rule of Fig. 1(a).

The following new lemma will serve to predict entering edges created by complete closure at a node.

Lemma 3. *Let v be an inner node of a balanced tree B. Let $e_1 = (v, u)$ and $e_2 = (v, w)$ be two consecutive edges around v in the clockwise order. During the closure algorithm, no edges will be inserted between e_1 and e_2 if and only if:*

(a) w is a leaf of B, or

(b) w is an inner node of B and the node t such that the edge $e_3 = (w, t)$ is the next edge around w after e_2 in the clockwise order is a leaf of B.

Proof. Let v an inner node of a balanced tree B. Let us consider two consecutive edges (v, u), (v, w) around v in the clockwise order. If w is a leaf, then during the closure it will merge with a node w' and close a triangular face enclosing the corner between (v, u), (v, w). No other edge can thus arrive at this corner. Assume now that w is an inner node of B. Let (w, t) be the next edge around w in the clockwise order. If t is a leaf of B then it will merge with u to form a triangular face and again no edge can arrive in the corner between (v, u), (v, w). In the other cases, (v, w) is an inner edge followed by another inner edge (w, t). Since an edge that forming a triangular face that encloses the corner between (v, u), (v, w) must from w, the corner is not enclosed. But at the end of the partial closure, there are no more pairs of consecutive inner edges: some edge must be arrived in the corner. □

Lemma 4. *Let $R = (T_0, T_1, T_2)$ be the minimal realizer encoded by a balanced tree B. A node v of B is a leaf of T_2 if and only if v has no incoming edge colored 2 in B and,*

1. *the parent edge of v in B is colored 2, or*

2. the parent edge of v in B is colored 1, or
3. the parent edge of v in B is colored 0 and v is the last child with an edge colored 0 in clockwise order around $P_B(v)$ and
   (a) the parent edge of $P_B(v)$ is colored 0, or
   (b) the parent edge of $P_B(v)$ is colored 2.

The number of vertices of B satisfying these conditions is denoted $\ell(B)$.

Proof. For the node v to be a leaf in T_2, it must have no incoming edge of color 2 in B, and no edge must be inserted between its outgoing edges of color 0 and 1. When the parent edge of v has color 2 or 1, the outgoing edge of color 0 connects to a leaf and Case (a) of the previous lemma ensures that no edge arrives between this outgoing edge of color 0 and the outgoing edge of color 1. When the parent edge of v has color 0, if the next edge in clockwise order around the parent $P_B(v)$ of v in B is an outgoing edge (of color 1), then Case (b) of the previous lemma ensures that no edge of color 2 arrives.

Finally we need to check in the remaining cases, an incoming edge of color 2 indeed arrives between the two outgoing edges of color 0 and 1. This could happen if the corner we consider was part of the unbounded face after the partial closure. But in the remaining cases, both the edge $(v, P_B(v))$ and the next edge in clockwise order around $P_B(v)$ are incoming. Since the form of the boundary after partial closure prohibits two consecutive incoming edges, the proof of the lemma is complete. □

From Lemma 4 and Theorem 2, we obtain:

Theorem 5. *Any well-orderly map with n nodes can be coded by a pair (B, W) where B is balanced tree of $\mathcal{B}_{n-2}$ and W a bit string of length $n + \ell(B)$. Encoding and decoding takes linear time.*

3.2 A context-free grammar for colored trees

We shall now give a recursive decomposition of trees in which the parameter ℓ of Lemma 4 can be followed.

To do this we consider the three sets $\mathcal{F}_i$, for $i = 0, 1, 2$ of trees with a root edge of color i.To a tree T of $\mathcal{F}_i$, $i = 1, 2$, we associate the parameter $k(T) = \ell(T)$. To a tree T of $\mathcal{F}_0$ we associate the parameter $k(T)$ defined as $\ell(T)$ except for the root node which contributes to $k(T)$ as soon as it has no incoming edge of color 2, and a second parameter $k'(T)$ defined as $\ell(T)$ except for the root node which never contributes.

The decomposition is obtained, classically, at the root node: a tree with root edge of color 0 is made of a root node that carries, in clockwise order, a sequence of subtrees of root color 1, an outgoing edge of color 2, a sequence of subtrees of root color 0, an outgoing edge of color 1, and a sequence of subtrees of root color 2. The parameter ℓ is almost additive on subtrees . However, due to Rule 3 in Lemma 4, the root of a subtree with root edge of color 0 may or may not be susceptible to contribute depending how it is attached. In other terms, depending of how it is attached, a subtree T' with root color 0 contributes $k(T')$ or $k'(T')$.

On Fig. 7 the decomposition is pictured schematically: an incoming edge represents a tree, a triangle represents a possibly empty sequences of subtrees, and color correspond to root colors. For color 0, plain and dashed lines respectively indicate positions where the contribution is given by parameters k or k'. Finally root nodes that contribute to the parameters are pictured in a box.

3.3 Generating functions of trees and the asymptotic number of well-orderly maps

The reader can refer to [GJ83] for a general presentation of enumeration of decomposable structures using grammars and generating series.

Let us consider the generating function $F_i(z, u)$ of trees with root color i, $i = 0, 1, 2$ with respect to the number of edges and the parameter k, and $F'_0(z, u)$ of trees with root color 0 with respect to the number of edges and the parameter k':

$$F_i \equiv F_i(z, u) = \sum_{T \in \mathcal{F}_i} z^{|T|} u^{k(T)} \quad \text{and} \quad F'_0 \equiv F'_0(z, u) = \sum_{T \in \mathcal{F}_0} z^{|T|} u^{k'(T)}.$$

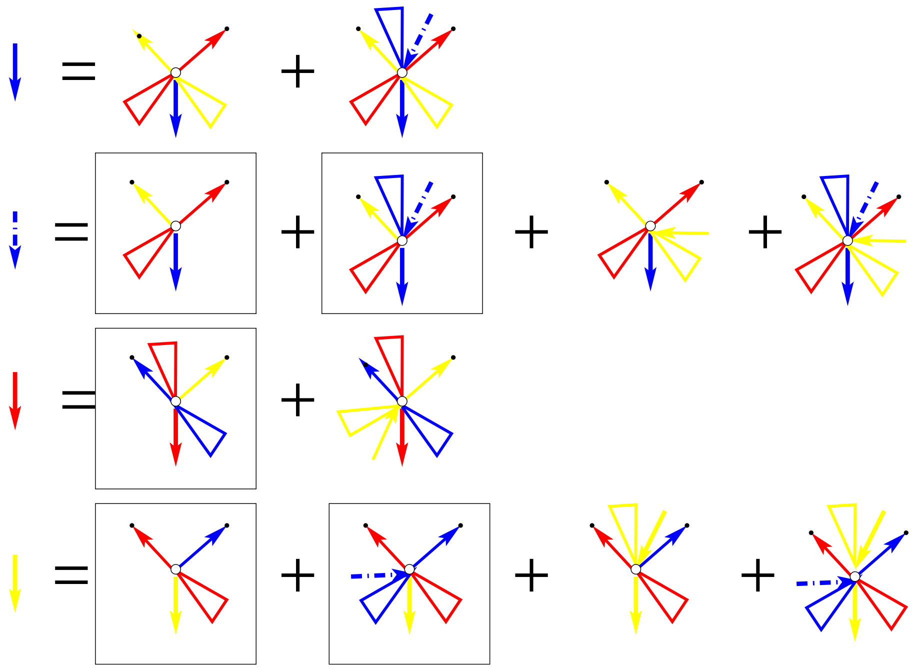

Fig. 7. A decomposition of colored trees allowing to track the contributions to ℓ.

Recall that with respect to additive parameters, the generating function of a possibly empty sequence of elements of a set S is the quasi-inverse $1/(1-f)$ of the generating function f of S. Therefore the previous decomposition translates into the following system of equations:

$$\begin{cases} F_0 = \dfrac{z\left(1+\frac{F_0'}{1-F_0}\right)}{(1-F_1)(1-F_2)}, \\ F_0' = \dfrac{z\left(u+\frac{F_2}{1-F_2}\right)\left(1+\frac{F_0'}{1-F_0}\right)}{1-F_1} \\ F_1 = \dfrac{z\left(u+\frac{F_2}{1-F_2}\right)}{(1-F_1)(1-F_0)}, \\ F_2 = \dfrac{z\left(u+\frac{F_2}{1-F_2}\right)\left(1+\frac{F_0'}{1-F_0}\right)}{1-F_1}, \end{cases} \quad \text{or} \quad \begin{cases} F_0 = \dfrac{z\left(1+\frac{F_2}{1-F_0}\right)}{(1-F_1)(1-F_2)}, \\ F_1 = \dfrac{z\left(u+\frac{F_2}{1-F_2}\right)}{(1-F_1)(1-F_0)}, \\ F_2 = \dfrac{z\left(u+\frac{F_2}{1-F_2}\right)\left(1+\frac{F_2}{1-F_0}\right)}{1-F_1}, \end{cases}$$

where the observation that $F_0'(z,u) = F_2(z,u)$ in the left hand side system yields the right hand side one. This system of equations completely defines the generating series $F_0(z,u)$. By elimination an algebraic equation for $F_0(z,u)$ is immediately obtained, which is of degree 4.

We are particularly interested in the case $u = 2$, since the coefficient f_n of z^n in

$$F(z) = F_0(z,2) = \sum_{T\in\mathcal{F}_0} z^{|T|}2^{\ell(T)},$$

counts n-node trees weighted by $2^{\ell(u)}$, and thus overcount n-nodes balanced trees with the same weight. According to Theorem 3, upon multiplying by 2^n, this yields an upper bound on the number of well-orderly maps with n nodes.

From elementary complex analysis, we have that $\log f_n \sim \log(\rho^{-n})$ where ρ is the radius of convergence of the series $F(z) = \sum_n f_n z^n$. Applying the implicit function theorem to the system defining $F(z)$, we can compute its radius of convergence and obtain:

$$\rho = \frac{\sqrt{189+114\sqrt{3}}-6\sqrt{3}-9}{4}.$$

From Theorem 5 we obtain:

Theorem 6. *The number of well-orderly maps with n nodes satisfies*

$$\frac{1}{n}\log_2 |H_n| \leqslant 1 + \log_2 1/\rho + o(1) \approx 4.9098 + o(1).$$

3.4 A code for colored trees

Let S be a binary string. We denote by $\#S$ the number of binary strings having the same length and the same number of ones than S. More precisely, if S is of length x and has y ones, then we set $\#S := \binom{x}{y}$.

Lemma 5. **[BGH03]** *Any binary string S of length n can be coded into a binary string of length $\log_2(\#S)+o(n)$. Moreover, knowing n, coding and decoding S can be done in linear time, assuming a* RAM *model of computation on $\Omega(\log n)$ bit words.*

Lemma 6. *Let B be a balanced tree such that the corresponding realizer $R = (T_0, T_1, T_2)$ has i_2 inner nodes in the tree T_2. The balanced tree B can be encoded with 5 binary strings S_1, S_2, S_3, S_4 and S_5 and 4 integers $a_0, a'_0, a_1, i_2 \leqslant n$ such that:*
$\#S_1 = \binom{n-a_0}{i_2-a_0}$, $\#S_2 = \binom{n-a_1}{a'_0}$, $\#S_3 = \binom{n+a_1}{a_1}$, $\#S_4 = \binom{a_1+a_0+a'_0}{a_0}$ *and* $\#S_5 = \binom{n-a_1-a'_0}{n-a_1-a'_0-i_2}$.

Lemma 7. *Let H be an m-edge well-orderly map. H can be encoded with 6 binary strings (*5 *for the minimal realizer and a last one to store the missing edges) and* 4 *integers $a_0, a_1, a'_0, i_2 \in [0, n]$ such that:* $\#S_1 = \binom{n-a_0}{i_2-a_0}$, $\#S_2 = \binom{n-a_1}{a'_0}$, $\#S_3 = \binom{n+a_1}{a_1}$, $\#S_4 = \binom{a_1+a_0+a'_0}{a_0}$, $\#S_5 = \binom{n-a_1-a'_0}{n-a_1-a'_0-i_2}$, $\#S_6 = \binom{2n-i2}{m-n-i_2}$.

Proof. With $S_1 - S_5$ a minimal realizer is encoded (Lemma 6). The last string indicates the edges to delete in order to rebuild the well-orderly map: for each v, one is used to indicate if the edge $(v, p_2(v))$ has to be removed and for each leaf v of T_2, one bit is used to indicate if the edge $(v, p_1(v))$ has to be removed. □

4 Applications

In view of Theorems 2 and 6, the number of connected planar graphs is at most $2^{4.9098n}$. As shown in [BGH03], the numbers of connected and general planar graphs differ by at most a polynomial term in n.

Theorem 7. *The number $p(n)$ of unlabeled planar graphs on n nodes satisfies, for every n large enough:*

$$\log_2 p(n) \leqslant \alpha n + O(\log n) \quad \textit{with} \quad \alpha \approx 4.9098.$$

This result is completed by the (known) lower bound

$$\log_2 p(n) \geqslant \beta n + O(\log n), \quad \text{with} \quad \beta \approx 4.767.$$

The length of the coding of well-orderly map depends of the number of the edges of the well-orderly map.

The following two results are obtained from the analysis of the length of the code of Lemma 7. The length of this code depends on the number of edges of the well-orderly map (see Fig. 8).

Theorem 8. *Almost all unlabeled and almost all labeled planar graphs on n nodes have at least $1.85n$ edges and at most $2.44n$ edges. Moreover, the result holds also for unlabeled connected and labeled connected planar graphs.*

Theorem 9. *Every connected m-edge planar graph can be encoded in linear time with at most $2.82m + o(m)$ bits.*

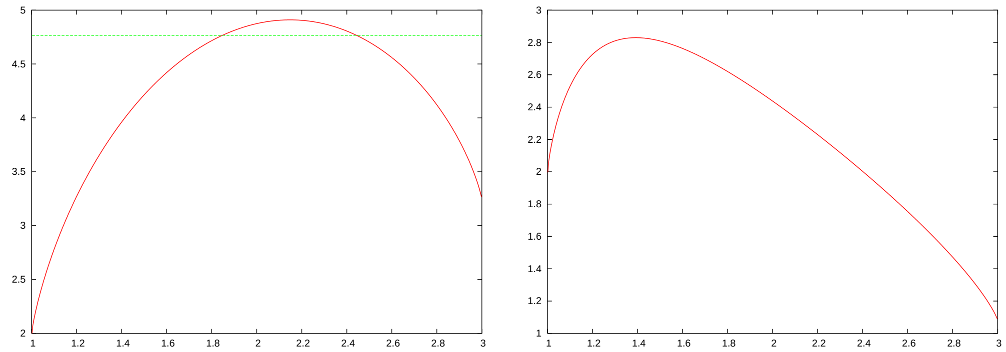

Fig. 8. (a) Number of bits necessary to encode a well-orderly map with $m = \alpha n$ edges, where $1 \leqslant \alpha \leqslant 3$. (b) Coding analyses: Number of bits per edges of a well-orderly map with $m = \alpha n$ edges, where $1 \leqslant \alpha \leqslant 3$.

5 The average size of planar drawings

Theorem 10. *The average number of leaves in a tree of a minimal realizer is* $5n/8 + o(n)$ *and the average number of 3-colored faces in a minimal realizer is* $n/8 + o(n)$.

Proof. Using classical techniques on generating function, we obtain that the average number of leaves of the tree T_0 of a minimal realizer is $5n/8 + o(n)$. By symmetry, this result is clearly true for the two other trees of the realizer. Since for any realizer, $\ell_0 + \ell_1 + \ell_2 + \Delta = 2n - 5$, where ℓ_i is the number of leaves in T_i and Δ is the number of 3-colored faces of the realizer [BLSM02b], the second result comes directely. $\square$

In [ZH03] a straight-line drawing algorithm base on minimal realizers is presented. This algorithm first computes the minimal realizer of a triangulation of the graph. Then the graph is drawn on a grid $(n - 1 - \Delta) \times (n - 1 - \Delta)$, where Δ is the number of 3-colored faces of the so obtained minimal realizer. Our analysis gives an average complexity of such drawings:

Corollary 1. *The average grid size required (i.e., the average width and the average height) to draw a triangulation is at most* $(\frac{7n}{8} + o(n)) \times (\frac{7n}{8} + o(n))$.

In [BLSM02a] a polyline drawing algorithm also based on minimal realizers is proposed. The graph is then drawn on a grid $(n - \lfloor \frac{\ell}{2} \rfloor - 1) \times \ell$, where ℓ is the number of leaves of the tree T_0 of the obtained minimal realizer $R = (T_0, T_1, T_2)$. Our analysis gives an average complexity of such drawings:

Corollary 2. *The average grid size required to draw a triangulation is at most* $(\frac{11n}{16} + o(n)) \times (\frac{5n}{8} + o(n))$.

References

[BGH03] Nicolas Bonichon, Cyril Gavoille, and Nicolas Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of Lecture Notes in Computer Science, pages 499–510. Springer, February 2003.

[BGK03] Manuel Bodirsky, Clemens Gröpl, and Mihyun Kang. Generating labeled planar graphs uniformly at random. In 30^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of LNCS, pages 1095–1107, 2003.

[BGW02] Edward A. Bender, Zhicheng Gao, and Nicholas C. Wormald. The number of labeled 2-connected planar graphs. *The Electronic Journal of Combinatorics*, 9(1):R43, 2002.

[BLSM02a] Nicolas Bonichon, Bertrand Le Saëc, and Mohamed Mosbah. Optimal area algorithm for planar polyline drawings. In 28^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 2573 of LNCS, pages 35–46. Springer, 2002.

[BLSM02b] Nicolas Bonichon, Bertrand Le Saëc, and Mohamed Mosbah. Wagner's theorem on realizers. In 29^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2380 of LNCS, pages 1043–1053. Springer, 2002.

[CGH⁺98] Richie Chih-Nan Chuang, Ashim Garg, Xin He, Ming-Yang Kao, and Hsueh-I Lu. Compact encodings of planar graphs via canonical orderings and multiple parentheses. In 25^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1443 of LNCS, pages 118–129. Springer, July 1998.

[CLL01] Yi-Ting Chiang, Ching-Chi Lin, and Hsueh-I Lu. Orderly spanning trees with applications to graph encoding and graph drawing. In 12^{th} *Symposium on Discrete Algorithms (SODA)*, pages 506–515. ACM-SIAM, January 2001.

[DVW96] Alain Denise, Marcio Vasconcellos, and Dominic J.A. Welsh. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.

[FJ89] Greg N. Frederickson and Ravi Janardan. Efficient message routing in planar networks. *SIAM Journal on Computing*, 18(4):843–857, August 1989.

[GH99] Cyril Gavoille and Nicolas Hanusse. Compact routing tables for graphs of bounded genus. In 26^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of LNCS, pages 351–360. Springer, July 1999.

[GJ83] I.P. Goulden and D.M. Jackson. *Combinatorial Enumeration*. John Wiley & Sons, 1983.

[GM02] Stefanie Gerke and Colin J.H. McDiarmid. On the number of edges in random planar graphs. *Combinatorics, Probability & Computing*, 2002. To appear.

[GN04] Omer Giménez and Marc Noy. Estimating the growth constant of labelled planar graphs. In 3^{rd} *Colloquium on Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities*. Birkhäuser, September 2004.

[KADS02] Andrei Khodakovsky, Pierre Alliez, Mathieu Desbrun, and Peter Schröder. Near-optimal connectivity encoding of 2-manifold polygon meshes. *Graphical Models*, 2002. To appear in a special issue.

[KR99] Davis King and Jarek Rossignac. Guaranteed 3.67V bit encoding of planar triangle graphs. In 11^{th} *Canadian Conference on Computational Geometry (CCCG)*, pages 146–149, August 1999.

[KW95] Kenneth Keeler and Jeffery Westbrook. Short encodings of planar graphs and maps. *Discrete Applied Mathematics*, 58:239–252, 1995.

[Lu02] Hsueh-I Lu. Improved compact routing tables for planar networks via orderly spanning trees. In 8^{th} *Annual International Computing & Combinatorics Conference (COCOON)*, volume 2387 of LNCS, pages 57–66. Springer, August 2002.

[LW87] Valer A. Liskovets and Timothy R. Walsh. Ten steps to counting planar graphs. *Congressus Numerantium*, 60:269–277, 1987.

[MR97] J. Ian Munro and Venkatesh Raman. Succinct representation of balanced parentheses, static trees and planar graphs. In 38^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 118–126. IEEE Computer Society Press, October 1997.

[MSW01] Colin J.H. McDiarmid, Angelika Steger, and Dominic J.A. Welsh. Random planar graphs, 2001. Preprint.

[PS03] Dominique Poulalhon and Gilles Schaeffer. Optimal coding and sampling of triangulations. In 30^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of LNCS, pages 1080–1094. Springer, July 2003.

[Ros99] Jarek Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):47–61, 1999.

[Sch90] Walter Schnyder. Embedding planar graphs on the grid. In 1^{st} *Symposium on Discrete Algorithms (SODA)*, pages 138–148. ACM-SIAM, 1990.

[Tho01] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. In 42^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, October 2001.

[Tur84] György Turán. Succinct representations of graphs. *Discrete Applied Mathematics*, 8:289–294, 1984.

[Wri71] E. M. Wright. Graphs on unlabelled nodes with a given number of edges. *Acta Math.*, 126:1–9, 1971.

[Yan89] Mihalis Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38:36–67, 1989.

[ZH03] Huaming Zhang and Xin He. Compact visibility representation and straight-line grid embedding of plane graphs. In *Workshop on Algorithms and Data Structures (WADS)*, volume 2748 of LNCS, pages 493–504. Springer, July 2003.