



Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs


Note

Local certification of graphs on surfaces [☆]

Louis Esperet ^{*}, Benjamin Lévêque

Laboratoire G-SCOP (CNRS, Univ. Grenoble Alpes), Grenoble, France



ARTICLE INFO

Article history:

Received 25 May 2021

Received in revised form 10 January 2022

Accepted 17 January 2022

Available online 20 January 2022

Communicated by A. Casteigts

Keywords:

Local certification

Proof labelling schemes

Planar graphs

Graphs on surfaces

ABSTRACT

A proof labelling scheme for a graph class \mathcal{C} is an assignment of certificates to the vertices of any graph in the class \mathcal{C} , such that upon reading its certificate and the certificates of its neighbors, every vertex from a graph $G \in \mathcal{C}$ accepts the instance, while if $G \notin \mathcal{C}$, for every possible assignment of certificates, at least one vertex rejects the instance. It was proved recently that for any fixed surface Σ , the class of graphs embeddable in Σ has a proof labelling scheme in which each vertex of an n -vertex graph receives a certificate of at most $O(\log n)$ bits. The proof is quite long and intricate and heavily relies on an earlier result for planar graphs. Here we give a very short proof for any surface. The main idea is to encode a rotation system locally, together with a spanning tree supporting the local computation of the genus via Euler's formula.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

The goal of local certification is to verify that a network, represented by a connected graph G in which each vertex has a unique identifier, satisfies some given property. The constraint is that each node of the network has a local view of the network (its neighborhood) and has to make its decision based only on this local view. If the graph satisfies the property, we want all vertices to accept the instance, while if the graph does not satisfy the property, at least one vertex has to reject the instance. This is a significant restriction and it only allows the verification of local properties (related to the degrees, for instance), so each vertex is given in addition some small certificate, and each vertex can now base its decision on its certificate and the certificates of its neighbors. For any property \mathcal{P} , the goal is to produce a protocol to certify \mathcal{P} locally while using certificates of minimal size. Such a protocol is called a *one-round proof labelling scheme with complexity $f(n)$* , where $f(n)$ is the maximum number of bits in the certificate of a vertex in an n -vertex graph satisfying \mathcal{P} (a formal definition of proof labelling schemes will be given in Section 2). Proof labelling schemes are a natural component of self-stabilizing algorithms, and are a particular form of distributed interactive protocols (with a single interaction). More broadly, proof labelling schemes with compact certificates (of logarithmic or polylogarithmic size) can be seen as a distributed version of the class NP, for which certificates of polynomial size exist (and can be checked in polynomial time in the centralized setting).

The Euler genus of a surface Σ is denoted by $\mathbf{eg}(\Sigma)$ (see Section 3 for more on surfaces and graph embeddings). In particular, the orientable surfaces of Euler genus 0, 2 and 4 are respectively the sphere (or equivalently the plane), the torus

[☆] The authors are partially supported by the French ANR Projects GATO (ANR-16-CE40-0009-01), GrR (ANR-18-CE40-0032), and by LabEx PERSYVAL-lab (ANR-11-LABX-0025).

^{*} Corresponding author.

E-mail addresses: louis.esperet@grenoble-inp.fr (L. Esperet), benjamin.leveque@grenoble-inp.fr (B. Lévêque).

and the double torus. The non-orientable surfaces of Euler genus 1 and 2 are respectively the projective plane and the Klein bottle.

Motivated by recent work on distributed interactive protocols in classes with linear time recognition algorithms [12], it was recently proved that graph planarity has a one-round proof labelling scheme with complexity $O(\log n)$ [6], and that this complexity is the best possible. More recently, the same authors built upon their previous work to extend their result to graphs embeddable on any fixed surface [7].

Theorem 1.1 ([7]). *For any (orientable or non-orientable) surface Σ , the class of graphs that are embeddable on Σ has a one-round proof labelling scheme with complexity at most $O(\sqrt{\mathbf{eg}(\Sigma)} \cdot \log n)$.*

The proof of the planar case (i.e., the case $\mathbf{eg}(\Sigma) = 0$) in [6] and its extension to general surfaces [7] are fairly intricate, with the two papers totaling 65 pages. The proof of the planar case [6] reduces the problem to graphs that are closer and closer to trees (for which compact proof labelling schemes are known), while the proof for general surfaces [7] works by carefully cutting the surface along non-contractible cycles, thus reducing the problem to planar graphs.

In this short note, we give a simple and direct proof of Theorem 1.1, based on rotation systems together with a distributed computation of the Euler genus using Euler's formula along a rooted spanning tree. We believe that our simplified approach is an important step towards an extension of this work to more general classes, such as minor-closed classes. In addition, we want to emphasize that surfaces are central in the study of distributed algorithms in planar graphs, as these graphs are locally indistinguishable from graphs on surfaces (see for instance [3] for applications of this observation to obtain lower bounds on distributed coloring of planar graphs).

Related work A reviewer pointed out an interesting article on a related topic, by Benjamini and Lovász [4], where it is proved that the local observation of some random process in an embedded graph allows to determine the genus of the embedding. The setting is quite different, as the random process is not distributed and the values taken by the edges are real numbers, but it might be the case that some ideas developed there can be useful in the field of local certification.

Organization of the paper The formal definition of proof labelling scheme is given in Section 2, and the basic terminology of graphs on surfaces is given in Section 3, along with a description of rotation systems and the Heffter-Edmonds-Ringel rotation principle. For the description of the certificates, we found it convenient to first present the orientable case (Section 4), which is slightly simpler, and then explain the small modifications we have to perform in the non-orientable case (Section 5). We could have presented everything in the latter setting, which is more general, but we believe it would have been harder to follow. We conclude with some open problems in Section 6.

2. Proof labelling schemes

All graphs in this paper are undirected, simple, and connected. A *one-round proof labelling scheme* for a graph class \mathcal{F} is a prover-verifier pair $(\mathcal{P}, \mathcal{V})$, with the following properties. For any integer $n \geq 1$ and any n -vertex graph $G \in \mathcal{F}$, whose vertices are assigned distinct identifiers $(\text{id}(v))_{v \in V(G)}$ from $\{1, \dots, \text{poly}(n)\}$, the *prover* \mathcal{P} assigns to each vertex $v \in V(G)$ a certificate $c_G(v) \in \{0, 1\}^*$ (that might depend on the vertex identifiers). The *verifier* \mathcal{V} satisfies the following properties for any graph G :

One-round: Each vertex $v \in V(G)$ collects the identifiers and certificates of its neighbors (*one-round*). Let $C_G(v) = (\text{id}(u), c_G(u))_{u \in E(G)}$.

Completeness: If $G \in \mathcal{F}$, then for any vertex $v \in V(G)$,

$$\mathcal{V}(\text{id}(v), c_G(v), C_G(v)) = 1.$$

Soundness: If $G \notin \mathcal{F}$, then for every possible choice of certificates $(c'_G(v))_{v \in V(G)}$ and distinct identifiers $(\text{id}(v))_{v \in V(G)}$, there exists a vertex $v \in V(G)$ such that

$$\mathcal{V}(\text{id}(v), c'_G(v), C'_G(v)) = 0,$$

where $C'_G(v) = (\text{id}(u), c'_G(u))_{u \in E(G)}$.

In other words, upon reading the identifiers and certificates of its neighbors and itself, each vertex of a graph $G \in \mathcal{F}$ *accepts* the instance, while if $G \notin \mathcal{F}$, for every possible choice of identifiers and certificates, at least one vertex *rejects* the instance.

The *complexity* of the labelling scheme is the maximum size $|c_G(v)|$ of a certificate in an n -vertex graph of \mathcal{F} . If we say that the complexity is $O(f(n))$, for some function f , the $O(\cdot)$ notation refers to $n \rightarrow \infty$. The definition above assumes that there is a single round of communication between the vertices (when each node collects the certificates of its neighbors), which is why this type of proof labelling scheme is called a *one-round proof labelling scheme*. There is a more general definition [8,10] in which each node is allowed to gather the certificates of its neighbors at distance t , for some integer $t \geq 1$, but in our case it is enough to restrict ourselves to $t = 1$.

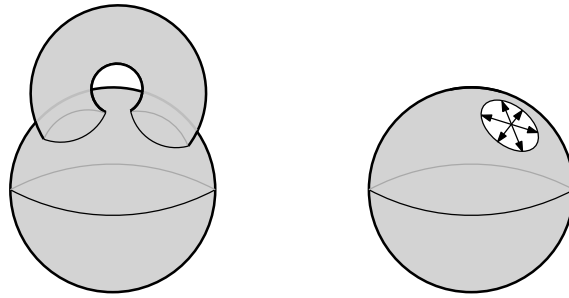


Fig. 1. A sphere with a handle (left) and a sphere with a cross-cap (right).

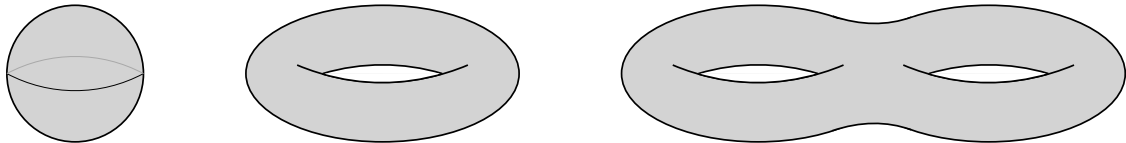


Fig. 2. From left to right: the sphere, the torus, and the double-torus.

3. Cellular embeddings and rotation systems

3.1. Surfaces

We refer the reader to the book by Mohar and Thomassen [11] for more details or any notion not defined here. A *surface* is a non-null compact connected 2-manifold without boundary (meaning that every point has a neighborhood that is homeomorphic to an open subset of the plane). By the classification theorem of surfaces, any surface is homeomorphic to one of the following (see Fig. 1):

- the *orientable surface of genus h* , obtained by adding $h \geq 0$ *handles* to the sphere.
- the *non-orientable surface of genus k* , obtained by adding $k \geq 1$ *cross-caps* to the sphere (where a *cross-cap* is a hole in the surface where antipodal points are identified).

The Euler genus of a surface Σ , denoted by $\mathbf{eg}(\Sigma)$, is equal to twice its genus if Σ is orientable, and as its genus otherwise. So, as already mentioned in the introduction the orientable surfaces of Euler genus 0, 2 and 4 are respectively the sphere (or plane), the torus and the double torus (see Fig. 2).

An *embedding* of a graph G on a surface Σ is a representation of G on Σ where the vertices of G are distinct elements of Σ , and every edge of G is a simple arc connecting in Σ the two vertices which it joins in G , with the property that the interior of every edge is disjoint from other edges and vertices. Removing the vertices and edges of G from the surface Σ divides Σ into connected components, called *faces* of the embedding of G (or faces of G , if the embedding is clear from the context). We say that an embedding is *cellular* if every face is homeomorphic to an open disk of \mathbb{R}^2 . If G is a graph with a cellular embedding in a surface of Euler genus g , then Euler’s formula states that

$$|V(G)| - |E(G)| + |F(G)| = 2 - g,$$

where $V(G)$, $E(G)$, and $F(G)$ denote the set of vertices, edges, and faces of (the embedding of) G .

In order to prove Theorem 1.1, it will be convenient to restrict ourselves to cellular embeddings. We will actually prove the following.

Theorem 3.1. *For any integer $g \geq 0$, the class of graphs with a cellular embedding on an orientable (resp. non-orientable) surface of Euler genus at most g has a one-round proof labelling scheme with complexity at most $O(\sqrt{g} \cdot \log n)$.*

Proof of Theorem 1.1 assuming Theorem 3.1. A graph is embeddable on an orientable surface Σ of Euler genus g if and only if it has a cellular embedding in an orientable surface of Euler genus at most g (see [11, Proposition 3.4.1]), so the orientable case of Theorem 1.1 follows directly from the orientable case of Theorem 3.1.

A graph G is embeddable on a non-orientable surface Σ of Euler genus g if and only if it has a cellular embedding in a non-orientable surface of Euler genus at most g or G is a tree (see [11, Proposition 3.4.2]). As trees have a simple one-round proof labelling scheme with complexity at most $O(\log n)$ [1], the non-orientable case of Theorem 1.1 can also be deduced directly from the non-orientable case of Theorem 3.1. \square

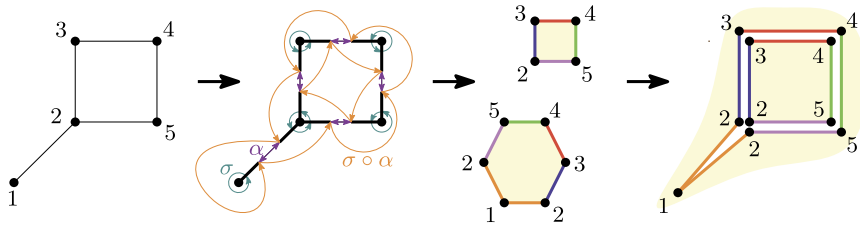


Fig. 3. The conversion of a graph embedded in the plane (or the sphere) into a rotation system, and back into a graph embedded in the sphere (obtained by gluing the two polygons on edges of matching colors). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

A graph G is said to be k -degenerate if there is an ordering v_1, \dots, v_n of the vertices of G , such that for any $1 \leq i \leq n$, the vertex v_i has at most k neighbors v_j with $j < i$. Note that the notion of degeneracy is very similar to the notion of *edge-arboricity*, which has been extensively studied in the context of distributed algorithms (the two parameters are within a multiplicative constant of each other). Using Euler’s formula, it is not difficult to derive the following result due to Heawood (see [11, Theorem 8.3.1]).

Theorem 3.2 (Heawood). For every $g \geq 0$, every graph embeddable on a surface of Euler genus at most g is k -degenerate, with $k = \max\left(5, \frac{5 + \sqrt{1 + 24g}}{2}\right)$.

In the next section we explain how the topological aspects of cellular embeddings can be translated into purely combinatorial notions in the case of orientable surfaces. The generalization to non-orientable surfaces is explained in Section 5.

3.2. Rotation systems

Let G be a graph. A *half-edge* of G is a pair (v, e) , where $v \in V(G)$ and e is an edge incident to v . We say that (v, e) and v are incident. The set of all half-edges of G is denoted by $B(G)$. A *rotation system* of G is a pair of permutations (σ, α) acting on $B(G)$, such that

- for any edge $e = uv \in E(G)$, $\alpha(v, e) = (u, e)$ and $\alpha(u, e) = (v, e)$ (i.e., α is an involution with no fixed point), and
- for each orbit of σ , there is a vertex $v \in V(G)$ such that the orbit consists of all the half-edges incident to v (in other words, we can view σ as a circular order on the half-edges incident to each vertex of G).

Each cellular embedding of a graph G in some orientable surface Σ can be translated into a rotation system by defining α as above and σ as the collection of circular orders on the half-edges around each vertex, in the positive orientation of the surface. Note that each orbit of $\sigma \circ \alpha$ corresponds to a different face of the embedding (where the half-edges appear in the negative orientation of the surface).

The Heffter-Edmonds-Ringel rotation principle (see Theorem 3.2.4 in [11]) states that every cellular embedding of a connected graph G in an orientable surface is uniquely determined, up to oriented homeomorphism, by its rotation system.

Although it will not be needed in the remainder, it is worth explaining how to recover the embedding of G in Σ from the rotation system (σ, α) with ground set B . Each orbit σ_v of σ is associated to a distinct vertex v , and each orbit α_e of α is associated to a distinct edge e connecting the two vertices associated to the two elements of α_e . The graph resulting from this construction is precisely G . To each orbit of $\sigma \circ \alpha$, we associate a polygon f whose sides are indexed by the edges of G and whose vertices are indexed by the vertices of G . Note that vertices and edges of G might appear several times on the same polygon or on different polygons (indeed, each edge appears twice and each vertex v appears $d(v)$ times among all the polygons). The circular order on the vertices and edges on each polygon in the negative orientation coincides with the circular order of the elements of B in the corresponding orbit of $\sigma \circ \alpha$. For any edge e of G , we glue the two polygons containing e together on e (if a single polygon contains e twice, we glue the two sides corresponding to e together), by respecting the natural orientation of e (that is, if $e = uv$, the vertex u of one polygon is identified with the vertex u of the other polygon, and similarly for v), see Fig. 3.

4. Certificates for orientable surfaces

In this section we prove the orientable case of Theorem 3.1. Recall that all graphs in this paper are assumed to be simple and connected.

Let G be a graph with a cellular embedding on an orientable surface Σ of Euler genus at most g . Let (σ, α) be the rotation system associated to some cellular embedding of G in Σ , and let $F(G)$ denote the set of faces of the embedding.

Our certificate consists in two parts: (1) a (local) description of the rotation system (σ, α) , and (2) a spanning tree of G which supports the (local) computation of the Euler genus, via Euler’s formula. Recall that according to Euler’s formula,

the value of g can be deduced from the values of the number of vertices, edges, and faces of the embedding. In order to compute the number of faces we will also need to store (locally) a circular order on the edges of each face. This will allow us to choose a root edge or vertex on each face, and avoid multiple counting of the same face along our spanning tree.

Edge certificates To describe our certificates it will be convenient to assume in the remainder that not only vertices, but also edges are given certificates, and that each vertex collects the certificates of its neighboring vertices and incident edges before choosing whether to accept or reject the instance.

Assume that G is a k -degenerate graph (with an ordering of its vertices witnessing this property), and each vertex v and edge e is assigned a certificate $c(v)$ or $c(e)$ of size at most t . Then each vertex u can store the certificates of the (at most k) edges uv such that v lies before u in the order. We are now in the case where only vertices store certificates (of size at most $t + k(t + \log n)$), and after collecting the new certificates of its neighbors, each vertex has access to the certificates $c(v)$ of its neighbors v and $c(e)$ of its incident edges e . Consequently, it follows from Theorem 3.2 that in order to prove Theorem 1.1, it is enough to assign certificates of size $O(\log n)$ to the vertices and edges of G (and to assume that each vertex collects the certificates of its adjacent vertices and incident edges before accepting or rejecting the instance). The same observation was used previously in [6,7] (see also [5]).

4.1. A distributed rotation system

Let $v \in V(G)$. The certificate of v contains the identifier $\text{id}(v)$ of v . Since for any vertex u in G , $\text{id}(u) \in \{1, \dots, \text{poly}(n)\}$, storing a constant number of identifiers takes $O(\log n)$ bits.

In the remainder, it will be convenient to talk about the *identifier* $\text{id}((v, e))$ of a half-edge (v, e) , which we define as the pair $(\text{id}(v), \text{id}(u))$, where u is the endpoint of e distinct from v .

Certificates around the vertices For each vertex v , let $(v, e_0), (v, e_1), \dots, (v, e_{d(v)-1})$ be the half-edges incident to v in the positive orientation, starting with some arbitrary half-edge (v, e_0) incident to v . For each $0 \leq i \leq d(v) - 1$, we say that (v, e_i) has v -index i , where v -indices are understood modulo $d(v)$, and we denote this half-edge by $\langle v \rangle_i$. By extension, we also say that the endpoint of e_i distinct from v has v -index i , so that the circular ordering of the half-edges around v coincides with a circular ordering of the neighbors of v (this is possible here since we deal with simple graphs). Then each edge uv is given as certificate the identifiers of u and v , together with the u -index of v and the v -index of u .

Certificates around the faces Consider a half-edge (v, e) . The face f associated to the orbit of $\sigma \circ \alpha$ containing (v, e) is said to be the *face bounding the half-edge* (v, e) , and we say that (v, e) is bounded by f . The half-edge $(\sigma \circ \alpha)(v, e)$ is called the *next half-edge on f* with respect to (v, e) . Note that if an edge e of G is incident to a single face f , the two half-edges of e are bounded by f , while if e is incident to two distinct faces f_1, f_2 , one half-edge of e is bounded by f_1 and the other is bounded by f_2 .

For each face f of G , the prover considers an arbitrary half-edge bounded by f and sets it as the *root* of f (in the remainder, if the root of f is (v, e) , we say that f *points to v*). The prover then assigns integers to the half-edges bounded by f as follows: for any half-edge (v, e) bounded by f , the f -index of (v, e) is the smallest integer $i \geq 0$ such that $(v, e) = (\sigma \circ \alpha)^i(u, e_0)$, where (u, e_0) denotes the root half-edge of f . So the root half-edge of f has f -index 0, and the maximum f -index is $d(f) - 1$, where $d(f)$ denotes the degree of f (the number of edges in a boundary walk of f , where edges appearing twice in the walk are counted with repetition). Note that if some half-edge has f -index i , the next half-edge on f has f -index $i + 1$ if and only if it is different from the root half-edge of f .

Now, consider any edge uv , and let f_u be the face bounding the half-edge (u, uv) and let f_v be the face bounding the half-edge (v, uv) . Then the edge uv is also given as certificate the f_u -index of (u, uv) together with the identifier of the root half-edge of f_u , and similarly the f_v -index of (v, uv) together with the identifier of the root half-edge of f_v .

Each vertex has degree at most $n - 1$ and each face has degree at most $2|E(G)| = O(n^2)$, so storing each v -index or f -index takes at most $O(\log n)$ bits. It follows that storing all the information described above takes at most $O(\log n)$ bits per vertex and edge, and thus $O(k \log n)$ per vertex (where $k = \max\left(5, \frac{5 + \sqrt{1 + 24g}}{2}\right)$, see Theorem 3.2).

We now describe the verification process at each vertex.

Verifying the vertices Each vertex collects the certificates of all its neighbors and incident edges. After having collected these certificates, both u and v are supposed to have all the *information concerning the edge uv* , namely: the identifiers $\text{id}(u)$ and $\text{id}(v)$, the u -index i of v , the v -index j of u , the identifiers of the root half-edges of the faces f and f' bounding $\langle u \rangle_i$ and $\langle v \rangle_j$ respectively, the f -index of $\langle u \rangle_i$ and the f' -index of $\langle v \rangle_j$. The verifier at each vertex v checks that the set of v -indices of the neighbors of v forms a circular permutation of $\{0, \dots, d(v) - 1\}$, thus certifying that the information collected by v is consistent with the local view of v in some embedding of G .

Let (σ, α) be the rotation system given by the v -indices of incident half-edges at each vertex v . By the Heffter-Edmonds-Ringel rotation principle, (σ, α) defines a unique cellular embedding of G on an orientable surface Σ (up to oriented homeomorphism). Note that for any edge uv where i is the u -index of v and j is the v -index of u , and f is the face bounding $\langle u \rangle_i$ in Σ , the next half-edge on f with respect to $\langle u \rangle_i$ is $\langle v \rangle_{j+1}$ (see Fig. 4).

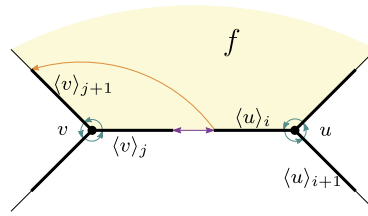


Fig. 4. The next half-edge on f .

Verifying the faces For each vertex v , and each $0 \leq j \leq d(v) - 1$ we do the following. Let u be the neighbor of v with v -index j , and let i be the v -index of u (see Fig. 4). The verifier at v checks that the half-edges $\langle u \rangle_i$ and $\langle v \rangle_{j+1}$ agree on the identifier of the root half-edge of the face f bounding them, so that the knowledge of the root half-edge of f is consistent along the face f . Since this verification is done by all vertices on the boundary of f , if no vertex rejects the instance, then each face has a unique root half-edge. In order to make sure that this root half-edge of f is actually bounded by f , the verifier at v simply checks that the f -index of $\langle v \rangle_{j+1}$ is equal to 0 if $\langle v \rangle_{j+1}$ is the root half-edge of f , or equal to 1 plus the f -index of $\langle u \rangle_i$ otherwise. Since the face f is finite and circularly ordered, some half-edge (u, e) bounded by f must have f -index at least the f -index of the next half-edge (u', e') on f , and by definition this is only possible if (u', e') has f -index 0. It follows that if the verifier at each vertex agrees with the instance, each face f has a unique root half-edge, and this root half-edge is bounded by f (so f points to a unique vertex, and this vertex is lying on f).

Hence, if no vertex has rejected the instance so far, each face bounds a unique root half-edge, and each vertex knows whether each of its half-edges is the root half-edge of the face bounding it. In particular, each vertex knows the number of faces pointing to it.

4.2. Computation of the Euler genus

Using the information collected by each vertex v , and assuming all vertices have accepted the instance so far, we now certify that the surface Σ has Euler genus at most g . To do this, it suffices to compute $|V(G)|$, $|E(G)|$, and $|F(G)|$ and apply Euler's formula. We will do so by collecting the number of vertices, edges and faces along a spanning tree. Let T be a rooted spanning tree in G with root r . This spanning tree is certified locally using the following classical scheme (see [1,2,9]): the prover gives the identifier $\text{id}(r)$ of the root of T to each vertex v of G , as well as $d_T(v, r)$, its distance to r in T , and each vertex v distinct from the root is also given the identifier of its parent $p(v)$ in T . The verifier at v starts by checking that v agrees with all its neighbors in G with the identity of the root r of T . If so, if $v \neq r$, v checks that $d_T(v, r) = d_T(p(v), r) + 1$. Once the rooted spanning tree T has been certified, each vertex of G knows its children in T . This can be used to check that

- all vertices agree on the same number $n = |V(G)|$ of vertices: In order to do this, the prover gives n to each vertex v of G , as well as a counter $\nu(v)$ which is equal to the number of vertices in the subtree of T rooted in v . The verifier at every vertex v simply checks that v has the same value of n as its neighbors in G , and that $\nu(v)$ is equal to 1 plus the sum of $\nu(u)$, for all children u of v (if any). Note that this can be checked locally. It only remains to check that for the root r of T , $\nu(r) = n$.
- all vertices agree on the same number $m = |E(G)|$ of edges: Again, the prover gives the value of m to each vertex v of G , together with a counter $\mu(v)$ defined as the half of the sum of the degrees $d_G(u)$ of the vertices u in the subtree of T rooted in v . The verifier at v only needs to check that it agrees on the value of m with its neighbors in G , and that $\mu(v)$ is $\frac{1}{2}d_G(v)$ plus the sum of $\mu(u)$, for all children u of v (if any). Since $m = |E(G)| = \frac{1}{2} \sum_{v \in V(G)} d_G(v)$, it remains to check that for the root r of T , $\mu(r) = m$.
- all vertices agree on the same number $|F(G)|$ of faces: Again, the prover gives the value of $|F(G)|$ to each vertex v of G , together with a counter $\phi(v)$ equal to the number of faces pointing to vertices lying in the subtree of T rooted in v (recall that each face has a unique root half-edge, and each vertex knows the number of faces pointing to it). The verifier at v checks that $\phi(v)$ is the number of faces pointing to v plus the sum of $\phi(u)$, for all children u of v (if any). It remains to check that for the root r of T , $\phi(r) = |F(G)|$.

It follows that, assuming no vertex has rejected the instance so far, each vertex has now access to $|V(G)|$, $|E(G)|$, and $|F(G)|$, and can check whether

$$2 + |E(G)| - |V(G)| - |F(G)| \leq g.$$

As a consequence of Euler's formula, this is equivalent to saying that the rotation system associated to G embeds G in an orientable surface of Euler genus at most g .

This concludes the proof of Theorem 1.1 for orientable surfaces.

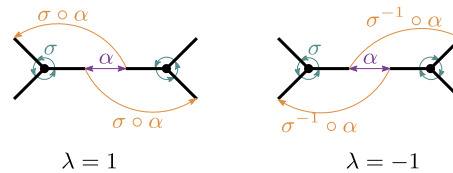


Fig. 5. Description of the next half-edge on a face depending whether the local orderings of two adjacent vertices are consistent (left) or not (right).

5. Non-orientable surfaces

In this section we explain how to prove the non-orientable case of Theorem 3.1. The case of non-orientable surfaces is very similar to the case of orientable surfaces, but there is an additional twist. An *embedding scheme* is a rotation system (σ, α) , except that each orbit e of α has a sign $\lambda_e \in \{-1, 1\}$. Given a cellular embedding of a graph G in a surface Σ (which is orientable or non-orientable), we can associate a circular order on the half-edges incident to each vertex v , by choosing an arbitrary orientation of the topological neighborhood of v (positive or negative). This choice of local orders around the vertices gives σ , and the edges give α , as before. Since we have chosen arbitrary orientations around the vertices, the orientations around two adjacent vertices u and v may not be consistent (i.e., agree on a small topological neighborhood around the edge uv). If they are consistent we set $\lambda_{uv} = 1$ and otherwise we set $\lambda_{uv} = -1$. The surface is orientable if and only if there is a choice of local orientations that is globally consistent, that is such that the resulting signs satisfy $\lambda_e = 1$ for every edge e . The surface is non-orientable if and only if G contains a cycle C which has odd number of edges e with $\lambda_e = -1$ (see Section 3.3 in [11]).

It turns out that the Heffter-Edmonds-Ringel rotation principle still holds in this more general setting (see Theorem 3.1.1 in [11]). The only difference when retrieving the faces of the embedding is that in order to find the next half-edge on some face, with respect to some half-edge (v, e) , we consider the sign λ_e . If $\lambda_e = 1$, the next half-edge on the face bounding (v, e) is $(\sigma \circ \alpha)((v, e))$, as before. If $\lambda_e = -1$, the next half-edge on the face bounding (v, e) is $(\sigma^{-1} \circ \alpha)((v, e))$. So instead of identifying the faces in the embedding of G with orbits of $\sigma \circ \alpha$ as before, we identify them with orbits of the function $\varphi : (v, e) \mapsto (\sigma^{\lambda_e} \circ \alpha)((v, e))$. This is illustrated in Fig. 5.

To adapt the certificate of the orientable setting to this more general framework, the value of λ_{uv} is added to the certificate of each edge uv . Using this additional information, the next half-edge on a face is computed using φ instead of $\sigma \circ \alpha$. A single bit is added to the certificate of every edge, and so at most $k = O(\sqrt{g})$ bits are added to the certificate of every vertex.

At this point, assuming no vertex has rejected the instance so far, we have certified that G can be embedded in some surface of Euler genus at most g (orientable or not). To conclude, it remains to certify that G can be embedded in a *non-orientable* surface of Euler genus at most g (we thank an anonymous referee for pointing out that this final step was missing in an earlier version of this manuscript). As suggested above, this can be done by certifying the existence of a cycle C which has an odd number of *negative* edges, i.e. edges e with $\lambda_e = -1$. To do so, the prover chooses the rooted spanning tree T of Section 4.2 such that

- its root r is in a cycle C with an odd number of negative edges,
- there is a negative edge e_r of C that is incident to r and not in T , and
- the path $C \setminus e_r$ is a path in T .

Then each vertex v stores the parity of the number of negative edges on the path from v to r in T . This can be certified by similar techniques as in Section 4.2, namely with a counter η at each vertex such that $\eta(r) = 0$ for the root r , and, for each vertex $v \in V(G) \setminus r$, the value of $\eta(v)$ is equal to $\eta(p(v)) + 1$ if $\lambda_{vp(v)} = -1$ and $\eta(p(v))$ otherwise. Finally the verifier at r checks that the value $\eta(v)$ of its neighbor v along e_r is equal to 0. With the additional edge e_r , this gives an odd number of negative edges in C , ensuring that the surface is non-orientable.

At most $O(\log n)$ bits are added to the certificates of the vertices to certify C , so the complexity of the proof labelling scheme remains $O(\sqrt{g} \log n)$.

6. Conclusion and open problems

In this paper we gave a simple proof of Theorem 1.1, which shows that embeddability in a fixed surface (and in particular planarity) has a one-round proof labelling scheme with complexity $O(\log n)$, which is best possible. Graphs embeddable on a fixed surface form important examples of *minor-closed classes*, that is classes \mathcal{F} such that for any graph $G \in \mathcal{F}$, any minor of G (i.e., any graph obtained from G by deleting vertices and edges and contracting edges) lies in \mathcal{F} .

The authors of [6,7] asked whether any minor-closed class \mathcal{F} has a one-round proof labelling scheme with complexity $O(\log n)$, and noted that even giving local certificates of $O(n^\alpha)$ bits for such classes, for some $\alpha < 1$, seems to be a difficult challenge. It can be observed that on the other hand, it follows from the Graph Minor Theorem of Robertson and Seymour [13] that for any proper minor-closed class \mathcal{F} , there is a one-round proof labelling scheme with complexity $O(\log n)$

for the *complement* of \mathcal{F} . This is because such a class \mathcal{F} has a finite number of minimal obstructions, and these obstructions (if they appear) can be easily certified using $O(\log n)$ bits per vertex (see [6], where constructions are given for specific minors). Note that compact proof labelling schemes for H -minor free graphs (when $|V(H)| \leq 4$) were given in [5].

We note here that even proving that graphs of bounded treewidth have one-round proof labelling schemes with complexity $O(\log n)$ seems to be non-trivial.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank the reviewers for their comments and suggestions.

References

- [1] Y. Afek, S. Kutten, M. Yung, The local detection paradigm and its application to self-stabilization, *Theor. Comput. Sci.* 186 (1–2) (1997) 199–229.
- [2] B. Awerbuch, B. Patt-Shamir, G. Varghese, Self-stabilization by local checking and correction, in: 32nd Symposium on Foundations of Computer Science (FOCS), 1991, pp. 268–277.
- [3] P. Aboulker, M. Bonamy, N. Bousquet, L. Esperet, Distributed coloring in sparse graphs with fewer colors, *Electron. J. Comb.* 26 (4) (2019) P4.20.
- [4] I. Benjamini, L. Lovász, Global information from local observation, in: 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2002, pp. 701–710.
- [5] N. Bousquet, L. Feuilloley, T. Pierron, Local certification of graph decompositions and applications to minor-free classes, ArXiv e-print, arXiv:2108.00059, 2021.
- [6] L. Feuilloley, P. Fraigniaud, I. Rapaport, É. Rémila, P. Montealegre, I. Todinca, Compact distributed certification of planar graphs, in: 39th ACM Symposium on Principles of Distributed Computing (PODC), 2020.
- [7] L. Feuilloley, P. Fraigniaud, P. Montealegre, I. Rapaport, É. Rémila, I. Todinca, Local certification of graphs with bounded genus, ArXiv e-print, arXiv:2007.08084, 2020.
- [8] M. Göös, J. Suomela, Locally checkable proofs in distributed computing, *Theory Comput.* 12 (1) (2016) 1–33.
- [9] G. Itkis, L.A. Levin, Fast and lean self-stabilizing asynchronous protocols, in: 35th Annual Symposium on Foundations of Computer Science (FOCS), 1994, pp. 226–239.
- [10] A. Korman, S. Kutten, D. Peleg, Proof labeling schemes, *Distrib. Comput.* 22 (4) (2010) 215–233.
- [11] B. Mohar, C. Thomassen, *Graphs on Surfaces*, Johns Hopkins University Press, Baltimore, 2001.
- [12] M. Naor, M. Parter, E. Yogev, The power of distributed verifiers in interactive proofs, in: 31st ACM-SIAM Symposium on Discrete Algorithms (SODA), 2020, pp. 1096–1115.
- [13] N. Robertson, P.D. Seymour, Graph minors. XX. Wagner's conjecture, *J. Comb. Theory, Ser. B* 92 (2) (2004) 325–357.