Algorithms and Combinatorics for Geometric Graphs (GeomGraphs)

Homework #2, due ~~November 26th~~ November 28th, 2025, AoE

This homework is due ~~November 26th~~ November 28th AoE (Anywhere on Earth). Please send me your solutions (or your questions) for this homework exclusively by email at arnaud.de-mesmay@univ-eiffel.fr. You can write either in French or in English.

This homework consists of one long exercise and one smaller, more open-ended exercise. They are independent and can be treated in any order. You can skip a question and use its results in the next questions if you explicitly say so.

This homework is optional and can only contribute positively to your grade. So do not censor yourself and do submit your homework even if it is partial and incomplete.

Exercise 1:

Throughout this exercise we work with orientable surfaces of genus $g > 0$. Recall that a **triangulation** of a surface is a cellularly embedded graph where every face has degree exactly three. We say that a triangulation is **k-cute** is it has the following two additional properties:

- All the vertices have degree at least k, and
- The graph dual to the triangulation is bipartite, i.e., the triangles can be colored black and white so that adjacent triangles have different colors.

Q1. Show that the torus admits no 8-cute triangulation.

Q2. Show that an 8-cute triangulation has $O(g)$ vertices.

Q3. Show that every orientable surface of genus $g \geq 4$ admits an 8-cute triangulation.[1] *Hint: It is easier to do it with $O(1)$ vertices.*

We now fix a 6-cute triangulation of a surface of genus $g \geq 2$. A **walk** on such a triangulation is a finite word $w = v_1 e_1 v_2 e_2 \dots e_{n-1} v_n$ where each edge e_i for $i \in [|1, n-1|]$ has endpoints the vertices v_i and v_{i+1}. A walk is **trivial** if $n = 1$ and **closed** if $v_k = v_1$. Let w be a walk and $e_{i-1} v_i e_i$ be a vertex of that walk surrounded by two incident edges. We say that w makes a **k-turn** at v_i for $k \geq 0$ if w leaves exactly k triangles to its left when going from e_{i-1} to e_i through v. So if w makes a 0-turn at v_i, then $e_{i-1} = e_i$, and if w makes a 1-turn, then e_{i-1} and e_i are consecutive in the clockwise ordering around v_i. Similarly, w makes a **$-k$-turn** at v_i for $k \geq 0$ if w leaves exactly k triangles to its right when going from e_{i-1} to e_i through v. Note that a 0-turn is the same as a -0-turn.

[1] If you find 8-cute triangulations for surfaces of genus 2 and 3, do not panic: they do exist but I just phrased the exercise that way because the general construction I have in mind only works for $g \geq 4$.

We say that a turn is **sharp** if it is a 0-turn, a 1-turn or a -1-turn, and **colorblind** if it is a 2-turn or a -2-turn where e_{i-1} is adjacent to a white triangle on its left and e_i is adjacent to a black triangle on its left. Beware that this definition is not as symmetric as it looks: colorblind turns always look at triangles on the left while -2-turns are defined with respect to triangles on the right. A walk is **well-behaved** if it admits neither sharp nor colorblind turns.

Q4. Show that if a walk starts with a 2-turn, then continues with an arbitrary number of 3-turns and finishes with a 2-turn, then exactly one of the two 2 turns is colorblind.

Q5. Let w be a closed walk so that S cut along w consists of exactly two components, and the component to the left of w is a topological disk D. So in particular w is simple, i.e., it has no repeated vertices nor edges. Denote by n_k the number of vertices on w which lie on the boundary of the disk and are incident to exactly k triangles of D. Show that $2n_1 + n_2 \geq 6 + \sum_{k \geq 4} n_k$.

Q6. Deduce from the two previous questions that there are at least three vertices in the walk w that make a sharp or a colorblind turn.

A walk can naturally be considered as a curve on a surface, i.e., a map $w : [0,1] \to S$. Two walks w_1 and w_2 with the same endpoints u and v are **homotopic** if there exists a continuous map $h : [0,1] \times [0,1] \to S$ so that $h(\cdot, 0) = w_1$, $h(\cdot, 1) = w_2$ and for all t in $[0,1]$, $h(0,t) = u$ and $h(1,t) = v$. Thus a homotopy is a continuous deformation between w_1 and w_2 which preserves their endpoints. The point of the exercise is the following two questions which are a bit harder than the previous ones.

Q7*. Show that in a 6-cute triangulation, any two homotopic well-behaved walks are equal.

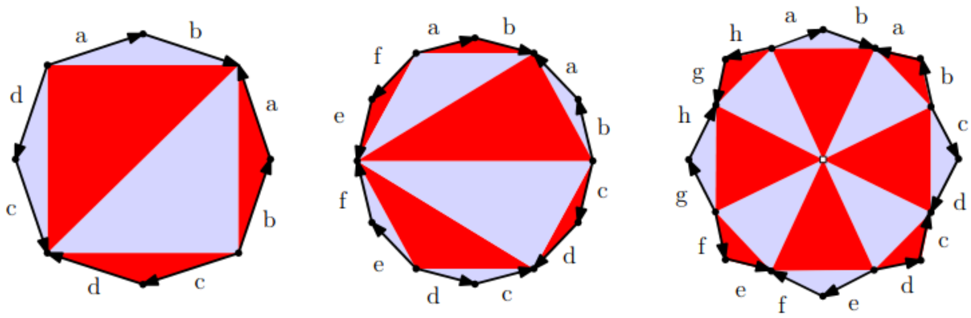
Q8*. Show that in a 6-cute triangulation, any walk is homotopic to a unique well-behaved walk. Provide a polynomial-time algorithm to compute it. *Hint: You can first try some homotopies to locally circumvent sharp and colorblind turns and see where it leads you.*

Solution: I obfuscated the names to make it a bit harder to google (and my few tries with ChatGPT suggested that it couldn't find it either). These are actually called k-reducing triangulations and are the state-of-the-art tool to test homotopy of curves or more complicated objects (e.g., graphs). This exercise is mostly taken from Section 3 of this recent paper by Éric Colin de Verdière, Vincent Despré and Loïc Dubois, you are encouraged to consult this paper if you want more details or are not satisfied with my solutions.

Q1. We apply the discrete Gauss-Bonnet formula where we put angles $1/6$ ($\pi/3$ radians) everywhere. For a 8-cute triangulation, this leads to negative curvature on the vertices and zero curvature on the faces, so negative total curvature. This is impossible for a torus since it has Euler characteristic zero.

Q2. We apply the same formula and the same angles. Each vertex contributes at least $-1/3$ curvature and the Euler characteristic is $2 - 2g$, so there are at $O(g)$ vertices.

Q3. The basic idea is to start from a polygonal scheme, add a vertex in the middle and put edges to every edge on the boundary. This is very good in terms of degrees but not so good in terms of bipartition of the faces. A bit of tinkering leads to a solution that works as in the following pictures, taken from the paper cited above:



The general solution is on the right, and the left and middle picture show ad hoc solutions for $g = 2$ and $g = 3$.

Q4. Let's assume that the first 2-turn is not colorblind, thus the second edge sees a white triangle on its left. Every subsequent 3-turn sees 3 triangles on its left, and thus by an immediate induction the second edge also always sees a white triangle on its left. When we reach the last 2-turn, the first edge thus sees a white triangle on its left and the second edge sees a black triangle on its left: it is colorblind.

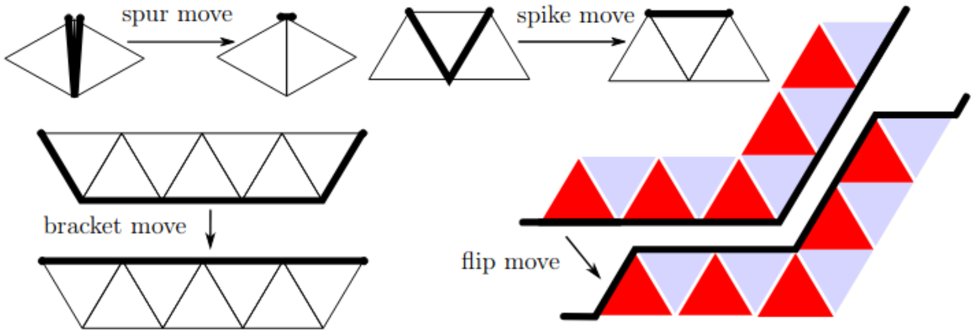
Q5. Note that there is no 0-turn by simplicity. We once again apply the discrete Gauss-Bonnet formula where we put angles 1/6 everywhere, this time in the disk D . As before, there is no curvature on the faces, nonpositive curvature at the inner vertices and curvature $(3 - k)/6$ at a boundary vertex incident to k triangles within the disk. This immediately yields the formula.

Q6. If $n_1 \geq 3$, the result is immediate. Otherwise, there is at least one 2-turn.

In that case, we split w in a set of maximal subwalks W starting and ending at a 2-turn and denote their numbers by ℓ (they overlap at the extremities). Out of those, let W' denote the walks consisting of one 2-turn, an arbitrary number of 3- or 1-turns and a last 2-turn (they might also overlap at their extremities), and denote their number by ℓ'. By **Q5**, each walk in W with a k-turn for $k > 3$ must be compensated by at least one 2-turn in a walk of W' or by half a 1-turn. So there are at least $n_2 - 2n_1$ 2-turns in walks in W'. By (the same argument as in) **Q4** these lead to at least $\lfloor n_2/2 \rfloor - n_1$ colorblind turns (the floor function takes into account consecutive subwalks sharing extremities, for example three consecutive 2-turns of walks in W' lead to 2 walks but only one colorblind turn). In total, there are at least n_1 sharp turns and $\lfloor n_2/2 \rfloor$ colorblind turns, thus at least 3 of either by **Q5**.

Q7. The proof is similar to that of the contractibility testing algorithm in the lecture notes. If two walks w_1 and w_2 are homotopic, their concatenation w is contractible. We lift in the universal cover. If either walk is not simple, it contains a closed simple subwalk which bounds a disk (since every simple closed walk is contractible and thus bounds a disk in the universal cover). Otherwise we can find disjoint subwalks w'_1 and w'_2, respectively of w_1 and w_2, such that $w'_1 \cup w'_2$ bounds a disk: indeed it suffices to consider a maximal subwalk w'_1 of w_1 that does not intersect w_2. In both cases, we obtain a disk bounded by either one or two subwalks, and thus having one or two distinguished vertices (the endpoint of the single walk or the common endpoints of the two walks). By **Q6** there are at least 3 sharp or colorblind turns, so one of them happens at a non-distinguished vertex. This contradicts the fact that w_1 and w_2 are well-behaved.

Q8. The uniqueness follows from the previous question, so we just have to prove existence. We provide local reduction rules that transform walks into well-behaved walks. These are illustrated in the following figure, also taken from the aforementioned paper. We only describe positive turns, the rules for the negative turns being identical.



Sharp turns are easily dealt with: 0-turns means that the walk makes a U-turn at an edge and thus could not have taken it, and whenever a walk makes a 1-turn, one can replace the two edges making this sharp turn by the opposite edge of the triangle. It is a homotopy, and it reduces the length of the walk, so after applying a finite number of these reduction rules there are no sharp turns anymore.

Colorblind turns can likewise be reduced by taking the opposite two edges in the rectangle defined by the two adjacent triangles, but one needs to be careful because this does not reduce the length, and might create new colorblind turns or sharp turn. Such a new colorblind turn is created by this reduction exactly when a colorblind 2-turn was adjacent to a 3-turn. This may happen both sides. Likewise, a new sharp turn is created when a colorblind 2-turn is adjacent to a 2-turn. In both cases, one should continue reducing until all the colorblind turns have been removed: leads to the flip move (when there are adjacent 3-turns which are inductively removed until there is none) or to the bracket move (when there is an adjacent 2-turn which becomes a 1-turn) pictured in the figure. Since each of these two moves removes at least one colorblind turn, after a finite number of applications there are no colorblind turns anymore.

The reduction rules are readily algorithmic since they amount to detecting anomalous turns in the turn sequence, and applying local modifications of length up to $O(|w|)$ (for the bracket and flip moves). This leads to a polynomial-time algorithm, which can actually be made linear with some tricks (see the paper, section 3.4).

As a side-note, ChatGPT 5 more or less aced the first six questions and then tried to confuse me by claiming that Q7 and Q8 are trivial because these are so-called CAT(0) spaces which are well-known to have unique geodesics. But these are not CAT(0) spaces, actually this is the reason why we need the clever bicoloring of the faces.

Exercise 2:

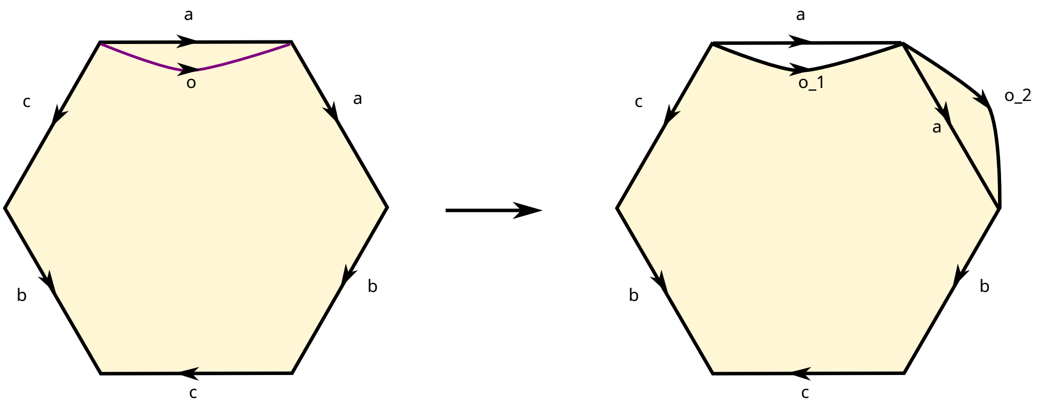
We say that a simple closed curve γ on a non-orientable surface S of genus $g \geq 1$ is **correcting** if cutting S along γ yields exactly one connected component, and that component is an orientable surface with boundary. Let G be a graph that is cellularly embedded on a non-orientable surface of genus $g \geq 1$. Provide a polynomial-time that computes a correcting curve in general position with respect to G. There are bonus points if you prove an upper bound[2] on the number of intersections of the curve with G in terms of the number of edges of G, and even more bonus points if you prove that your upper bound is optimal.

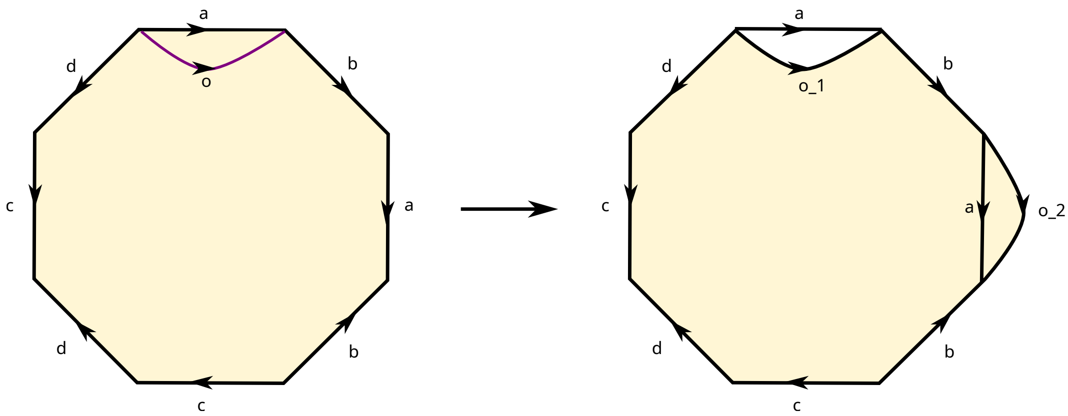
[2]The lower the better but do not bother with the constants and focus on the asymptotics $O(f(|E(G)|)$.

Solution: Here again I obfuscated the name but there is actually no consistent name in the literature for these curves. I like **orienting** which I have used in a few of my papers. The exercise is inspired from this paper where they are called **orientation-enabling curves**. The optimal upper bound in terms of length is provided in Proposition 5.2 there. A sketch of that proof is in Proposition 2.7 of this paper of mine.

Non-orientable surfaces have multiple equivalent characterizations, and the exercise can be solved in many ways depending on which one you take.

Solution 1:





By the classification of surfaces, every non-orientable surface admits a polygonal scheme $a_1a_1 \dots a_ga_g$. If g is odd, it also admits a polygonal scheme

$$aaa_1b_1\overline{a_1}\overline{b_1}\dots a_{(g-1)/2}b_{(g-1)/2}\overline{a_{(g-1)/2}}\overline{b_{(g-1)/2}},$$

as you can verify by yourself by just checking that this is indeed a non-orientable surface of genus g. Now if we cut along a curve o parallel to the a curve, we obtain a new polygonal scheme

$$o_1o_2a_1b_1\overline{a_1}\overline{b_1}\dots a_{(g-1)/2}b_{(g-1)/2}\overline{a_{(g-1)/2}}\overline{b_{(g-1)/2}},$$

where o_1 and o_2 are edges that are not attached to anything. See the above picture, top. This is an orientable surface where o_1 and o_2 form one boundary.

If g is even, we work with the polygonal scheme

$$aba\overline{b}a_1b_1\overline{a_1}\overline{b_1}\dots a_{(g-2)/2}b_{(g-2)/2}\overline{a_{(g-2)/2}}\overline{b_{(g-2)/2}},$$

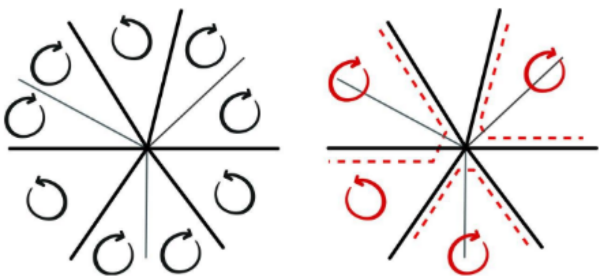
which you can also verify by yourself is a non-orientable surface of genus g. Then we cut once again along a curve close to the a curve, as in the above picture, right, and we obtain the polygonal scheme

$$o_1 b o_2 \overline{b}\overline{a_1}\overline{b_1} \dots a_{(g-2)/2} b_{(g-2)/2} \overline{a_{(g-2)/2}}\overline{b_{(g-2)/2}},$$

which is an orientable surface with two boundaries[3]. In both these cases the curve might not be in general position with respect to the graph, but any small deformation (homotopy) will do. Now in order to get a polynomial-time algorithm, it suffices to observe that the proof of the classification of surfaces I gave in class is algorithmic and can be implemented in polynomial-time (these are just local reduction rules on words and there is a clear measure of progress at each step). The bound on the number of edges is not so easy to track through the reductions though, and even if you do track it, it will not be optimal, essentially because this solution is overkill for our problem. This leads us to the second solution.

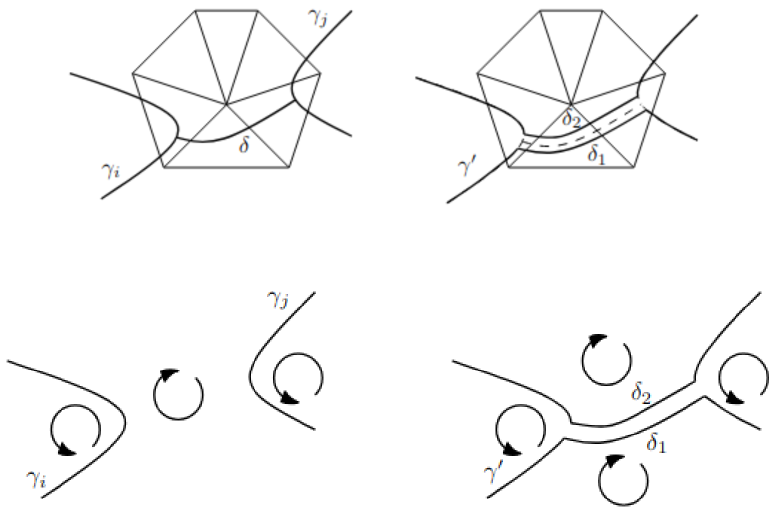
Solution 2: A surface is non-orientable if and only if there is no consistent orientation of the faces (Proposition 2.5 in the lecture notes): if we choose any choice of orientation of clockwise or counter-clockwise orientation of the faces and orient the incident edges accordingly, there will be an edge between two adjacent faces with the same orientation. This might be a bit confusing initially but think of an edge between two adjacent triangles which are both oriented clockwise: the edge will inherit opposite orientations from each triangle. So the discrepancy is really when the orientation given by both faces is the same. This is what happens when we see a letter twice with the same orientation on a polygonal scheme.

So one can start with an arbitrary orientation of the faces and mark each edge where the two incident orientations match. Around each vertex, we see an even number of such orientation matches (because they happen at a local change of orientation, and there is an even number of such local changes of orientation around a vertex). So the marked edges form an Eulerian subgraph (every vertex has even degree), which we can perturb a bit and transform into a family of curves as in the picture below.



The resulting family of curves $\mathcal{C}$ cuts the surface into an orientable surface (since there are no orientation mismatches anymore). But we want a single curve. In order to do so, we can inductively dig tunnels connecting pairs of curves in $\mathcal{C}$. These tunnels are computed by taking any spanning tree of the graph and doubling the edges connecting two different curves in $\mathcal{C}$, and moving them a bit aside so that they are in general position with respect to G. This leads to a single curve, and it is orienting: in order to see this one can subdivide the triangulation and orient the triangles consistently, see pictures below (taken from the aforementioned article of Matoušek, Sedgwick, Tancer and Wagner).

[3]One can show that when g is odd, any orienting curve is one-sided (i.e., when one follows it on the right, one ends up on the left after one turn, as the center curve of a Möbius band), while when g is even, any orienting curve is two-sided, see Lemma 5.3 in the first paper I cited. This is what leads to one vs two number of boundaries after cutting: cutting along a one-sided curve always yields a surface with a single boundary of length twice that curve (try it at home with a Möbius band in paper or watch this video).



By construction, the orienting curve crosses each edge $O(1)$ times and the process we have used to build it is a polynomial-time (actually linear-time) algorithm. This is tight: if you define a surface by taking a long cylinder of diameter $\Theta(|E|)$ and gluing a Möbius strip at each extremity, any orienting curve will have to hit both Möbius strips (otherwise the surface obtained after cutting would contain one of them and would thus be non-orientable) and thus be of length $\Omega(|E|)$.

As a side-note when I gave this exercise to ChatGPT 5 it got the algorithm right (although phrased in fancier terms) but was adamant that one could get an $O(g)$ bound.