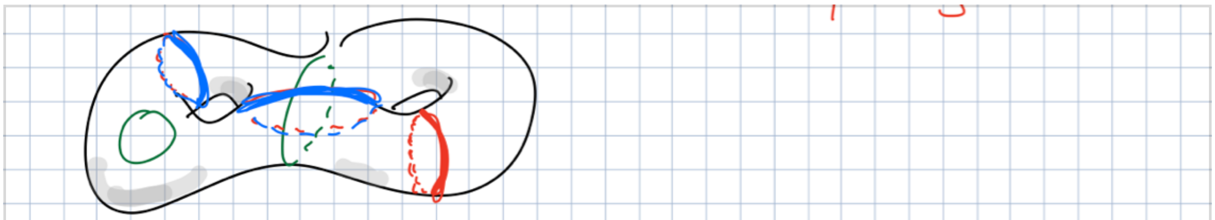
CS 7301.003.20F Lecture 16–October 12, 2020

Main topics are #minimum_cut and #homology.

Minimum Cuts and Homology

- A few weeks ago, we discussed an $O(n \log n)$ time algorithm for computing minimum s,t-cuts in planar graphs. Today, we'll extend that algorithm to work in low genus surfaces. We'll assume the surfaces are orientable for today.
- Remember in the plane, the minimum s,t-edge cut was dual to a cycle separating s^* and t^*.
- But when you have genus, you can have dual cycles that don't separate anything!
- And sometimes, the minimum cut may even consist of up to $g + 1$ dual cycles.

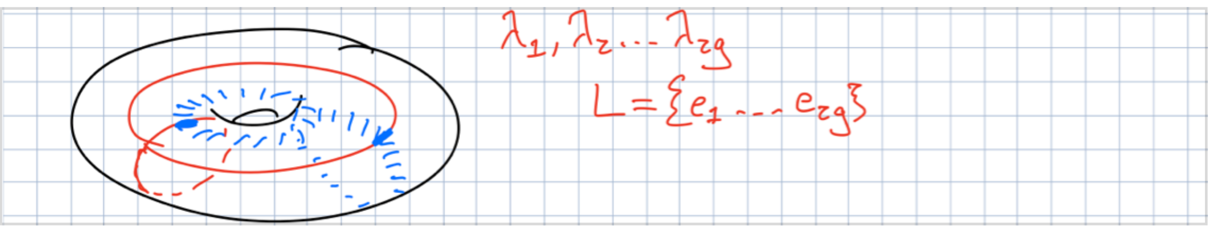


- So we need some way of characterizing groups of cycles to know if they separate s from t.
- Recall an *even subgraph* is a subgraph with all degrees even.
- Equivalently, it's the symmetric difference of several simple cycles's edges. Also, the symmetric difference of two even subgraphs is always an even subgraph.
- In fact, they form a vector space $Z_1(G) = Z_2^{E - V + 1}$.
  - Take any spanning tree T.
  - The $E - V + 1$ fundamental cycles are independent, because they all contain a distinct edge of $E \setminus T$.
  - But if you take any subset E' of $E \setminus T$ and ask for the even subgraph using exactly those edges of $E \setminus T$, you have no freedom over which edges of T to use.
  - So those fundamental cycles form a basis.
- We need some way of characterizing how even subgraphs lie on the surface. Homotopy is too restrictive, though, and doesn't really make as much sense when an even subgraph can have multiple components.
- Define a *boundary subgraph* as the boundary of a union of faces. If the surface itself has boundary, our subset can only contain faces that aren't marked "gone".
- Given a vertex v, every face shares an even number of edge incidences with v. Therefore, every boundary subgraph is even.
- Also, taking the symmetric difference of two boundaries gives us another boundary. We have another vector space $B_1(G) = Z_2^{F - 1}$.
  - While there are 2^F subsets of faces, pairs of them have the same boundary.

    - In fact, we could pick an arbitrary face f_infty, require it to never be in a subgraph, and then get a bijection between the boundary and remaining 2^{F - 1} subsets of faces.
- We say two even subgraphs A and B are *homologous* if and only if A oplus B is boundary.
- Homology forms an equivalence relation, and we can define a vector space over the homology classes of even subgraphs where the sum operation is taking the symmetric difference of representative members.
- In particular, the *homology classes* form the *homology space* H_1(G) = Z_1 / B_1 = Z_2^{V + E - F + 2} = Z_2^{2g}.
- If the surface has at least one boundary component, then B_1(G) = Z_2^{F - b}, meaning H_1(G) = Z_2^{2g + b - 1}. This expression 2g + b - 1 is sometimes referred to as the first Betti number, beta. We'll talk more about Betti numbers later.
- We'll assume from here on that there's always at least one boundary component.

Homology and Cohomology Bases

- To better understand the homology space, it's useful to find some representative cycles of the various homology classes.
- Let (T, C, L) be an arbitrary tree-coforest decomposition where T is a spanning tree, C^* is a spanning forest with one component per puncture, and L = {e_1, …, e_{2g + b - 1}} is a set of 2g + b - 1 leftover edges.
- Lemma: Gamma = {cycle(T, e) | e in L} = {gamma_1, …, gamma_{2g + b - 1}} forms a set of cycles in independent homology classes.
    - Take any subset of the cycles. I claim their symmetric difference is not a boundary.
    - In particular, take any cycle gamma_i in the subset, and let lambda_i be the unique cycle or puncture-to-puncture path in C^* + e_i. p_i crosses that symmetric difference of edges exactly once. But that means the edges don't form a boundary of present faces.
- We sometimes call Gamma itself a *homology basis*, because every even subgraph is homologous with some subset of Gamma.
- There's also a *cohomology basis* which kind of describes homology in the dual graph.
- Let Lambda = {lambda_1, …, lambda_{2g + b - 1}} be the set of cycles or puncture to puncture paths you get adding each e_i to C^*. Observe that Lambda only contains cycles if b = 1 or 0.

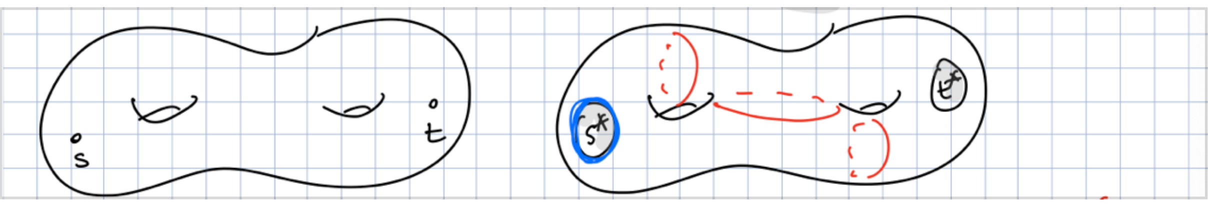


- Cohomology bases don't help us represent homology classes so much as know what class a subgraph belongs to.

- Define the *signature* [e] of an edge e as a vector of 2g + b - 1 bits where the ith bit is 1 if and only if e in lambda_i.
- The signature of a subgraph A is the sum of its edges' signature vectors mod 2.
- Lemma: Two subgraphs A and B are homologous if and only if [A] = [B] if and only if [A oplus B] = 0.
  - Every present face f is incident to exactly 0 or 2 edges of each path lambda_i.
  - Therefore, A oplus B has the same signature as any subgraph it is homologous to.
  - So if A oplus B is a boundary, then [A oplus B] = 0.
  - Suppose A oplus B is homologous with some non-empty subset of cycles from
    Gamma instead. [gamma_i] has its ith bit and only that bit set to 1, because it shares e_i with lambda_i and no other edges of C^* or L. So any non-empty subset of Gamma has non-zero signature and so does A oplus B.

Minimum Cut and Minimum Homologous Subgraph

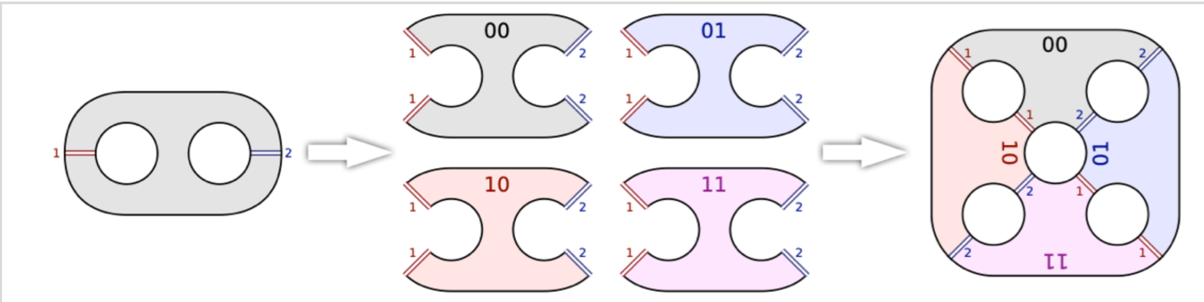
- So back to minimum cut.
- Like in the planar case, we're interested in finding a subgraph of the dual that separates s^ * from t^ *, meaning it should look like a boundary.
- If we remove faces s^* and t^*, then the minimum cut is no longer a boundary subgraph.
- But it is homologous to partial s, the boundary of s. In fact, the members of that homology class are precisely the dual s,t-edge cuts. The faces for the boundary you add onto partial s are dual to the vertices on the S side of the cut.



- So now we've reduced to the follow problem: Given an even subgraph eta on a surface graph with weighted edges, find a minimum weight subgraph eta' such that [eta] = [eta'].
- Bad news: we just reduced to an NP-hard problem. You can prove hardness using a reduction from max cut.
- But good news: the problem is *fixed parameter tractable* in the genus g. There's an f(g) * poly(n) time algorithm.
- So what are f(g) and poly(n)?
- We'll first solve a slightly simpler subproblem: find the shortest cycle gamma' such that [eta] = [gamma'].
- To do so, we'll treat this shortest cycle as a shortest path with the correct signature as we've done before. But how do we find the shortest path with the correct signature?
- We'll build something called the Z_2-homology cover Gbar = (Vbar, Ebar):

- Vbar = {(v, h) | v in V and h in Z^{2g + b - 1}}
- Ebar = {(u, h) (v, h') | uv in E and h oplus h' = [uv]}

- The idea here is that any path from some (v, 0) to (v, h) projects to a cycle containing v with signature [h] by just dropping the second component from the vertex labels.
- Each facial walk in G lifts to 2^{2g + b - 1} faces in the homology cover. Intuitively, what we've done is cut the surface along the dual paths in Lambda, made one copy of the surface per member of Z_2^{2g + b - 1}, and then glued them together again so the signature's of adjacent copies match which member of Lambda we're gluing along.
- Here's an example on the sphere with 3 boundary components (a pair of pants). Drawing the cover even for the torus is too messy to attempt.



- After some algebra, we can verify the cover has genus O(2^{2g + b - 1} (2g + b - 1)) and size O(2^{2g + b -1} n).
- So now what we could do is for each vertex v, find the shortest path from (v, 0) to (v, [eta]) and return the projection of the best path found. We'd spend like 2^O(g + b - 1) n^2 log n time.
- But that's not very helpful for minimum cut! Can we be any faster?
- Recall the forest-cotree decomposition from last week. Using it, we were able to make a cutgraph out of O(g + b) shortest paths and additional edges.
- The cut graph bounds a disk. Unless [eta] = 0 (in which case, we should just return the empty even subgraph), gamma' cannot bound a subset of faces. It must cross the cutgraph, touching one of its shortest paths sigma.
- In the Z_2-homology cover, each lift of sigma contains the first vertex of some lift of gamma'. sigma is a shortest path, so the lift of gamma' should stick to the lift of sigma before leaving and never coming back.
- We can safely cut open that lift of sigma turning it into a face. Then, run multiple-source shortest paths! We'll spend 2^O(g + b - 1) n log n time per choice of sigma for 2^O(g + b - 1) time total.
- Now going back to finding eta', which may have multiple cycles. We solve this problem using dynamic programming: For each k ≤ g + 1, find the best even subgraph of a particular signature that uses at most k cycles. We'll guess the signature of one its cycles and use the algorithm we just described to find the best cycle of that signature. Then, we'll use a recursively computed solution to find the other up to k - 1 cycles. Again, 2^O(g + b - 1) n log n time total (although that constant in the big-Oh is rather large).

- And since b = 2 for minimum s,t-cut, we get the minimum s,t-cut in $2^{O(g)} n \log n$ time.
  (It's really $64^g g^3 n \log n$, but hey, that's $O(n \log n)$ if g is constant!)