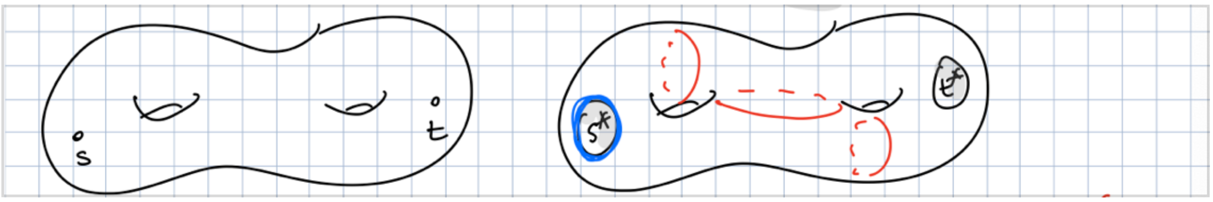
CS 7301.003.20F Lecture 17—October 14, 2020

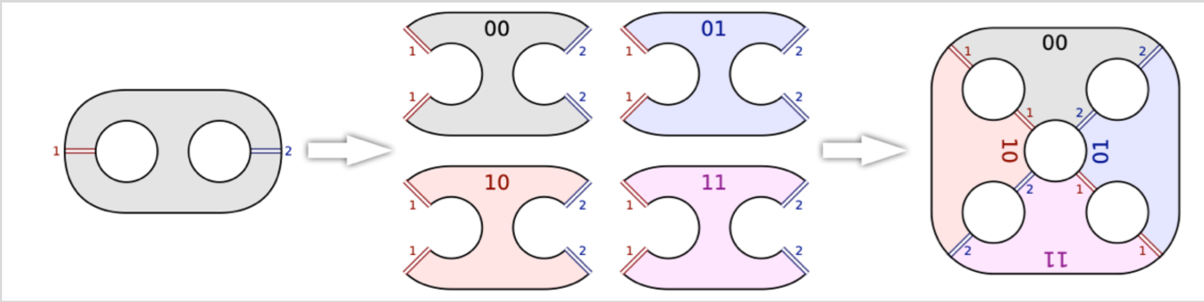
Main topics are #minimum_cut and #homology.

Minimum Cut and Minimum Homologous Subgraph

- Let's talk some more about 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 non-negatively weighted edges, find a minimum weight subgraph eta' such that [eta] = [eta'].
- 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) n log n 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!)
- This algorithm was found by Erickson and Nayyeri ['11].
- A different algorithm by Chambers et al. and Italiano et al. ['09, '11] solves the problem in g^O(g) n log log n time.

Global Minimum Cut

- Let's move on to a similar problem: *global minimum cuts*.
- The global minimum cut of a graph (sometimes just called the minimum cut) is the non-

empty edge cut of smallest total weight. It's also the cheapest of all s,t-cuts for all choices of s and t. Again, we're given an undirected graph with non-negative edge weights.

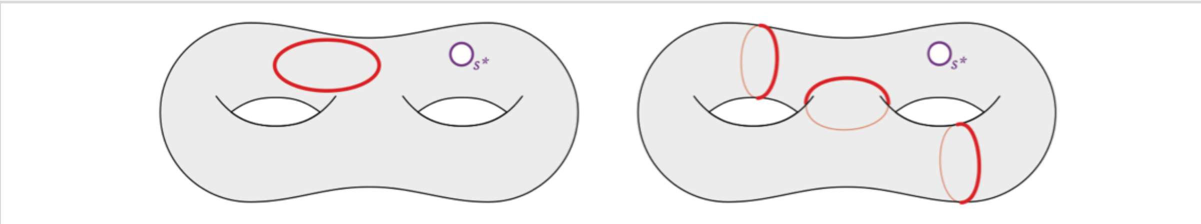
- In general graphs, we can solve the problem by just fixing some vertex s and computing the minimum s,t-cut for every choice of $t \neq s$. But that takes $O(n^2 m)$ time.
- There's also an $O(nm + n^2 \log n)$ time algorithm by Nagamochi and Ibaraki ['92], and if you allow randomness, Karger ['00] shows how to do it in only $O(m \log^3 n)$ time.
- But we can do better on a low genus surface and without the small chance of failure randomness provides. Let's see what happens in the plane first by considering an algorithm of Chalermsook et al. ['04]:
- So first, the minimum cut is a bond, and bonds are dual to cycles. So we must be looking for a minimum length cycle gamma in the dual graph. So let's stick to the dual graph.
- We're going to do a very simple divide-and-conquer strategy. First, pick any one vertex r, and let T_r be a shortest path tree out of r.
- Let cycle(T_r, pq) be a fundamental cycle separator. It partitions the plane into an inside and outside.
- There are three cases: if gamma lies entirely on the inside or outside of cycle(T_r, pq), we can find it by recursively searching in that component.
- Otherwise, it crosses the shortest path to p or q. But just like with minimum s,t-cut, we know it won't cross that path twice. So it crosses on of the shortest paths to go inside cycle(T_s, pq), and then it must touch the other path somewhere below the lowest common ancestor of p and q in T_r.
- Let s^* be the outside face incident to the first edge on the shortest path from r to p. Let t^* be the inside face incident to pq. gamma must separate s^* and t^*.
- And we already know an $O(n \log \log n)$ time algorithm to find such a cycle gamma!
- So we do two recursive calls and one $O(n \log \log n)$ minimum s,t-cut computation. We're using balanced separators, so there are $O(\log n)$ levels of recursion, and we spend a total of $O(n \log \log n)$ time on all subproblems in each level. The total running time is $O(n \log n \log \log n)$.
- Ła¸cki and Sankowski ['11] figured out how to play with dense distance graphs to improve the time to $O(n \log \log n)$.

On Surfaces

- So what if we're not in the plane but instead on a low genus orientable surface?
- We'll again go looking in the dual graph, but again we have the issue that not every cycle is dual to a cut and not every dual cut is a single cycle.
- Let s^* be an arbitrary face. We're going to focus on the problem of separating s^* from some non-empty subset of faces.
- To make it easier to describe such subsets, we'll cut s^* out of the surface to create one

boundary component. We're now looking for the cheapest non-empty boundary subgraph.

- Let's call the boundary of a non-empty set of faces a *separating subgraph*, meaning we're trying to find a minimum weight separating subgraph.
- We can't just look for the cheapest subgraph eta' s.t. [eta'] = 0 using our previous algorithm, though. We'd just get the empty boundary subgraph as the answer.
- Instead, let's look at some examples of what the solution eta may look like. There's two cases we need to consider.



- In the first case, eta is a contractible cycle.
    - The inside is a disk, just like when we solved the planar case. Maybe there's a way we can reduce to the planar case?
    - Take the greedy forest-cotree decomposition again, and build a reduced cutgraph from it. You can prove that each boundary-to-boundary path is shortest for its homotopy class.
    - And therefore, eta' never crosses it. In short, if a contractible cycle crosses the cutgraph, there must be a "bigon" where it crosses an arc and then turns around at the same arc. But then you could shortcut either the arc or the contractible cycle without changing either's homotopy class.
    - So, if we cut along the cutgraph, we get a planar disk containing eta'.
    - Now we just run Ła¸cki and Sankowski ['11]. Their algorithm will return a cycle separating some subset of faces F' on the inside from those on the outside. s^* is boundary, so it doesn't appear as part of F'. The boundary of F' in the original graph still separates F' from s^*, even if eta' isn't contractible.
        - But if eta' is contractible, then that's what we'll find.
- And what if eta' isn't contractible?
    - In this case, the subset of faces bounded by eta' don't form a disk. Therefore, their induced subgraph must have non-trivial homology classes.
    - But kind of like with the short contractible cycles, the cheapest members of these homology classes don't like crossing eta' if they can help it.
    - Lemma: Let gamma be cycle separated from s^* by eta' such that [gamma] ≠ 0. Let gamma' be the shortest subgraph such that [gamma] = [gamma']. Then gamma' lies on the opposite side of eta' from s^*.
    - The proof is another shortcutting argument, but made more difficult because eta' and gamma' could have multiple components.

    - Now, if we know the homology class of a such a gamma where [gamma] ≠ 0, then we can compute gamma' in g^O(g) n log log n time.
    - Every edge of gamma' is incident to at least one face separated from s^* by eta'. Let t_1^* and t_2^* be two faces incident to one of the edges of gamma'.
    - We then compute minimum s,t_1 and s,t_2-cuts. One of those will be the global minimum cut.
    - We don't actually know the homology class of gamma in advance, though, so we'll just try all 2^{2g} - 1 possibilities and return the best eta' we find.
- And we don't know if eta' is contractible or not, so we try both strategies and return the better subgraph we find.
- The whole algorithm takes g^O(g) n log log n time total.