CS7510 Graph Algorithms, Fall 2019	Lecture 9, Date Sep 24

Gomory-Hu Trees

Presenter: *Saurabh Sawlani* *Scriber:* Thomas Kim

Disclaimer: The notes are slightly different from the organization in class because, on introspection, this has a better flow.

Definition 0.1 (Minimum $s-t$ cut)**.** Given a graph $G(V, E, c)$, where c denotes positive edge capacities, and two vertices $s, t \in V$, the minimum $s-t$ cut is a partition of V into sets $A \ni s$ and $V \setminus A \ni t$ such that the sum of capacities of edges going across the partition is minimized.

We know that the minimum $s-t$ cut is also the maximum $s-t$ flow in the graph G, and hence one can find the value by one max-flow computation. This is currently $O(mn)$ time.

So, if we want to find the min $s-t$ cut for all s, t pairs, a naive way to do it is by running $\binom{n}{2}$ max-flow computations. But can we do better?

1 Min-cut has nice structural properties

Let us denote using $f_G(s, t)$, the min $s-t$ cut in G.

Claim 1.1. *For any 3 vertices $i, j, k \in V$, assume WLOG that $f_G(i, j) \leq f_G(j, k) \leq f_G(i, k)$. Then, $f_G(i, j) = f_G(j, k)$.*

Proof. □

The above claim hints that there are a lot of repeated values among the $\binom{n}{2}$ min-cut values. The following corollary follows directly from it.

Corollary 1.2. *For any 3 vertices $i, j, k \in V$,*

$$f_G(i, k) \geq \min(f_G(i, j), f_G(j, k))$$

Now consider a path between two vertices u and v: $u, w_1, w_2, \ldots, w_k, v$. Using the above corollary repeatedly, we get that

$$f_G(u, v) \geq \min\left(f_G(u, w_1), f_G(w_1, w_2), \ldots, f_G(w_k, v)\right). \quad (1)$$

Now, suppose we construct a complete graph H on the same vertex set V, where the cost of each edge (u, v) is $f_G(u, v)$. Let T be any maximal spanning tree of H. Consider an edge (u, v) which is not in the tree T. Let $u, w_1, w_2, \ldots, w_k, v$ be the tree path between u and v. Then, by the maximality of T

$$f_G(u, v) \geq \min\left(f_G(u, w_1), f_G(w_1, w_2), \ldots, f_G(w_k, v)\right).$$

Combining this with equation 1, we get that for any two vertices $u, v \in G$,

$$f_G(u, v) = \min_{e \in \mathcal{P}_T(u,v)} f_G(e),$$

where $\mathcal{P}_T(u, v)$ denotes the path in T between u and v. We call such a tree T a *Gomory-Hu tree* of G.

2 Computing a Gomory-Hu tree

We cannot compute H and find a spanning tree, because that would again be doing $\binom{n}{2}$ max-flow computations. We will, in fact, need exactly $n - 1$ max-flow computations.

To motivate the algorithm, consider the following instance. Let $A \ni s, \overline{A} \ni t$ represent a min $s - t$ cut. Let $u, v \in A$. Let G' represent the graph obtained by contracting $\overline{A}$. Then,

Claim 2.1. $f_G(u, v) = f_{G'}(u, v)$

Proof. Now, let $B \ni u, \overline{B} \ni v$ represent a min $u - v$ cut. Let $X := A \cap B$, $\hat{X} := A \cap \overline{B}$, $Y := \overline{A} \cap B$, $\hat{Y} := \overline{A} \cap \overline{B}$. Assume that $s \in X$ and $t \in Y$ (the other cases follow symmetrically).

Let $b_{UU'}$ denote the total capacity of edges crossing from some U to U'. Since $A, \overline{A}$ is a min s, t cut and $X \cup \hat{X} \cup \hat{Y}, Y$ is another s, t cut,

$$\begin{aligned} b_{A\overline{A}} &\leq b_{X \cup \hat{X} \cup \hat{Y}, Y} \\ b_{XY} + b_{X\hat{Y}} + b_{\hat{X}Y} + b_{\hat{X}\hat{Y}} &\leq b_{XY} + b_{\hat{X}Y} + b_{\hat{Y}Y} \\ \implies b_{X\hat{Y}} + b_{\hat{X}\hat{Y}} - b_{\hat{Y}Y} &\leq 0. \end{aligned}$$

Since $B, \overline{B}$ is a min u, v cut and $X \cup Y \cup \hat{Y}, \hat{X}$ is another u, v cut,

$$\begin{aligned} b_{B\overline{B}} &\leq b_{X \cup Y \cup \hat{Y}, \hat{X}} \\ b_{X\hat{X}} + b_{X\hat{Y}} + b_{\hat{X}Y} + b_{Y\hat{Y}} &\leq b_{X\hat{X}} + b_{\hat{X}Y} + b_{\hat{X}\hat{Y}} \\ \implies b_{X\hat{Y}} + b_{Y\hat{Y}} - b_{\hat{X}\hat{Y}} &\leq 0. \end{aligned}$$

Adding the two above equations, we get that $b_{X\hat{Y}} = 0$ (because it cannot be negative). We also get that $b_{Y\hat{Y}} = b_{\hat{X}\hat{Y}}$ and hence $b_{X \cup \hat{X} \cup \hat{Y}, Y} = b_{A\overline{A}}$.

Noting that $X \cup \hat{X} \cup \hat{Y}, Y$ is also an s, t cut in the contracted graph G', we get

$$b_{A\overline{A}} = f_G(u, v) \leq f_{G'}(u, v) \leq b_{X \cup \hat{X} \cup \hat{Y}, Y} = b_{A\overline{A}}.$$

This proves the claim. □

Algorithm

1. Set $V_T = V_G$, $E_T = \emptyset$
2. Choose some element X of V_T with more than 1 element.
3. Create a new graph G' by collapsing all elements of V_T other than X.
4. Choose 2 vertices u, v in X.
5. Compute the min cut in G' between u and v, say A, $\overline{A}$.
6. Split X into two blobs X_1 and X_2 corresponding to $X \cap A$ and $X \cap \overline{A}$. Add an edge of weight $f_{G'}(u, v)$ between X_1 and X_2. For any earlier edge $(Y, X) \in E_T$, if $Y \in A$, replace it with an edge Y, X_1, otherwise add Y, X_2 of the same weight.
7. Repeat this process until V_T contains only singletons.

Claim 2.2. *The above algorithm outputs a Gomory-Hu tree of G. In other words, for any two vertices $u, v \in V$*

$$f_G(u, v) = \min_{e \in \mathcal{P}_T(u,v)} w(e).$$

Proof. $f_G(u, v) \leq \min_{e \in \mathcal{P}_T(u,v)} w(e)$, because each of those edges in $\mathcal{P}_T(u, v)$ corresponds to a valid $u - v$ cut.

To show the other direction, consider at any point during the construction, suppose an edge of weight w connects two blobs X and Y. Then, there *must* exist vertices $x \in X$ and $y \in Y$ such that $f_G xy = w$.

This is trivially true in the beginning (no edges in T). Now assume that this statement is true until some step k in the algorithm. At step k, suppose we split a blob A into blobs A_1 and A_2. We know that the new edge (a_1, a_2) between A_1 and A_2 satisfies the property $f_G(a_1, a_2) = w(A_1, A_2)$. So we only need to verify that it is true for already existing edges. Suppose there was an edge in T between A and another blob B. By induction, let $a \in A$ and $b \in B$ satisfy $f_G(a, b) = w(A, B)$. Assume WLOG that B is now connected to A_1. If $a \in A_1$, then the property continues to hold. If $a \in A_2$, we need to show that there is another vertex $a' \in A_1$ such that $f_G(a', b) = w(A_1, B)$.

Using the triangle inequality,

$$f_G(b, a_1) \geq \min(f_G(b, a), f_G(a, a_2), f_G(a_2, a_1)).$$

From Claim 2.1, we know that collapsing A_2 does not affect the value of $f_G(b, a_1)$ and so we can eliminate the middle term (contracted edge can be assumed to have infinite capacity) Additionally, the new edge (a_1, a_2) also represents a cut between a and b and so $f_G(b, a) \leq f_G(a_2, a_1)$. This gives

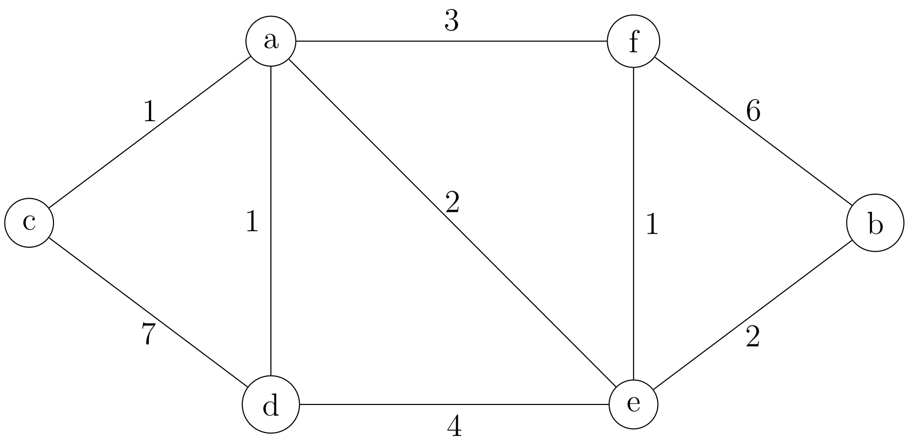
$$f_G(b, a_1) \geq f_G(b, a).$$

However, we know that there exists a cut of value $f_G(b, a)$ between b and a_1. So $f_G(b, a_1) = f_G(b, a)$.

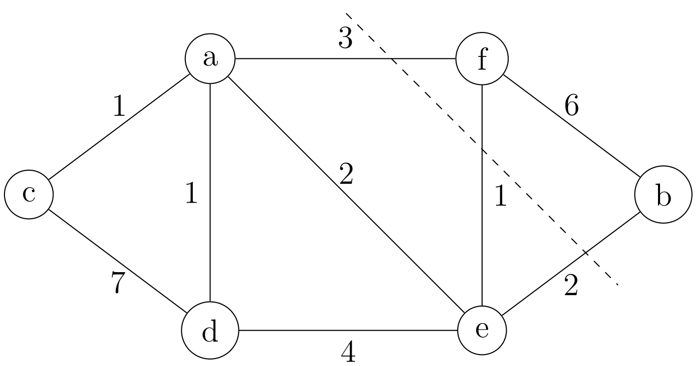
Now, using the fact that each edge represents a flow, the claim follows using the triangle inequality. □

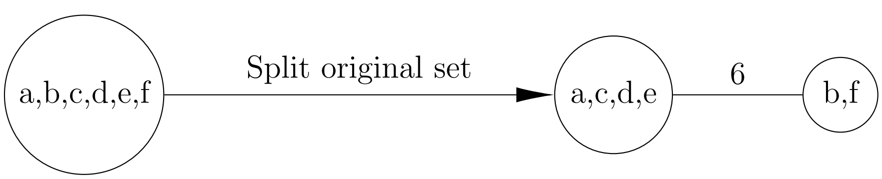
Gomory-Hu Tree Example

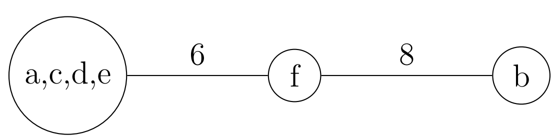
Initial graph:



1. Pick a, b (arbitrary). a-b min-cut = 6:

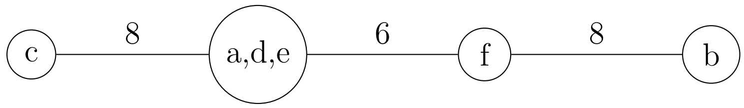




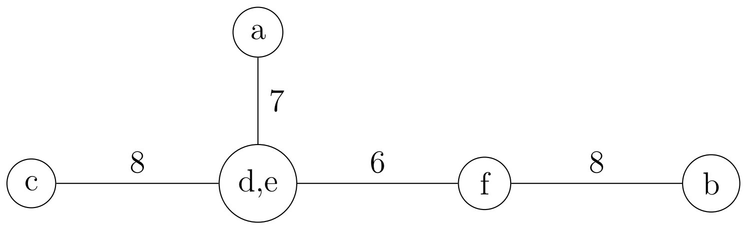


2. Pick b, f (again, arbitrary). b-f min-cut = 8

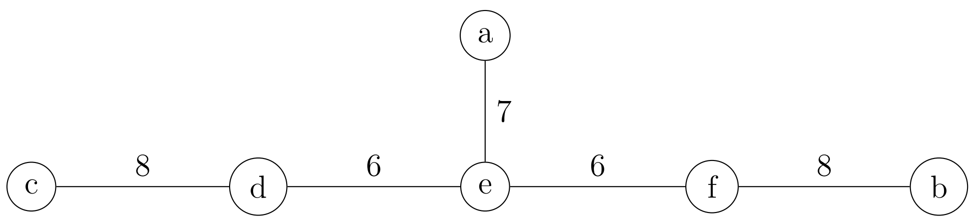
3. Pick c, d. c-d min-cut = 8



4. Pick a, e. a-e min-cut = 7



5. Pick d, e. d-e min-cut = 6



Gomory-Hu tree completed.