Gomory-Hu trees (from Korte & Vygen Section 8.6)

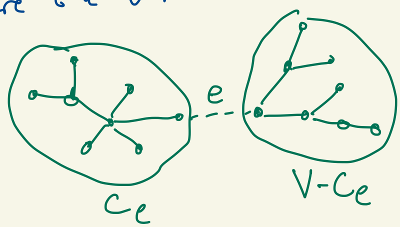
Definition 8.31 Let $G=(V,E)$ and $u: E \to \mathbb{R}_+$

A tree T is a Gomory-Hu tree for G if

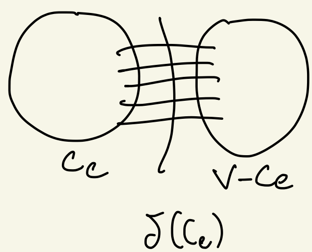
- $V(T) = V(G)$
- $\forall s,t \in V(G): \lambda_G(s,t) = \min\{u(\delta_G(C_e)) \mid e \in E(P_{st})\}$

Here P_{st} is the unique (s,t)-path in T and for all $e \in E(T)$

C_e, $V-C_e$ are the vertex sets of the 2 connected components of $T-e$



In G we denote by $\delta(C_e)$ the set of edges between C_e and $V-C_e$



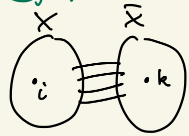
Goal: prove that every (G,u) has a Gomory-Hu tree

Consequence: $\forall (G,u) \exists$ list of $n-1$ cuts such that $\forall p,q \in V$ one of these cuts is a minimum (p,q)-cut

Lemma 8.30 $\forall i,j,k \in V(G)$ we have $\lambda_{ik} \geq \min\{\lambda_{ij}, \lambda_{jk}\}$

where λ_{ij} is the maximum number of edge-disjoint (i,j)-paths in G

proof: Consider a minimum (i,k)-cut $(X, \bar{X})$



$\lambda_{st} = u(\delta(X))$

if $j \in X$ then $\lambda_{jk} \leq \lambda_{ik}$

if $j \in \bar{X}$ then $\lambda_{ij} \leq \lambda_{ik}$ □

Main idea:

- Choose arbitrary $s,t \in V$ and find a min (s,t)-cut (A,B) $(\lambda_{st} = u(\delta(A)))$
- If $\max\{|A|,|B|\} \geq 2$ then may assume $|B| \geq 2$
- Contract A to a single vertex and denote the resulting graph by G/A
- Choose distinct vertices s', t' in B
- Find a minimum (s',t')-cut in G/A
- continue this process by always choosing new vertices s', t' that are not separated by any of the cuts we have found so far

$\boxed{\text{at each step, for every previously determined cut } (A', B') \text{ we contract one of } A', B' \text{ so that } s', t' \text{ are in the non contracted part}}$

The process ends when each pair of vertices is separated by at least one of the previously determined cuts (there is no non-contracted set of size ≥ 2)

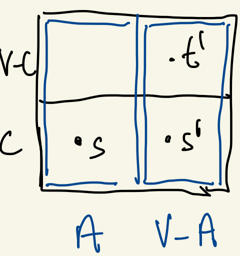
We shall prove that the $n-1$ cuts determined give us a Gomory-Hu tree T

Main observation: The value of a minimum (s',t')-cut in G/A is the same as the value of a minimum (s',t')-cut in G

Lemma 8.32 Let $G=(V,E)$ and $u: E \to \mathbb{R}_+$ be given.
Let $s,t \in V$ and let $(A, V-A)$ be a minimum (s,t)-cut wrt u.
Let $s',t' \in V-A$ and let G', u' be obtained by contracting A in G.
Then
$\forall$ minimum (s',t')-cut $(K \cup \{A\}, V'-K-\{A\})$ in G'
we have $(K \cup A, V - K \cup A)$ is a minimum (s',t')-cut in G
that is, $u(\delta(K \cup A)) = u'(K \cup \{A\})$ $\quad$ ($\square$)

Proof: It is sufficient to show that there exist a min (s',t')-cut $(A', V-A')$ in (G,u) such that $A \subseteq A'$ (contracting A will not affect such a cut)

Let $(C, V-C)$ be an arbitrary minimum (s',t') cut in (G,u) s.t. $s',t' \in V-A$ so the picture is as in the diagram below
without loss of generality we have $s \in C$



as the capacity function $u()$ is submodular, we get

$$u(\delta(A)) + u(\delta(C)) \geq u(\delta(A \cap C)) + u(\delta(A \cup C))$$
$$\geq u(\delta(A)) + u(\delta(A \cup C))$$

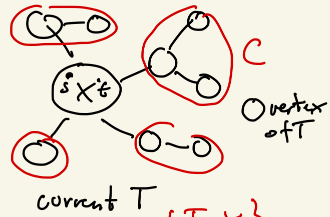
$(A \cap C, V - A \cap C)$ is an (s,t)-cut so $u(\delta(A \cap C)) \geq u(\delta(A))$

$\Rightarrow u(\delta(C)) \geq u(\delta(A \cup C))$ so $(A \cup C, V - A \cup C)$ is also a min (s',t')-cut in (G,u) $\Rightarrow$ $(C \cup \{A\}, V' - C - \{A\})$ is a min cut in (G', u') $\square$.

Gomory-Hu algorithm

(1) Initialize $V(T)=\{V(G)\}$, $E(T)=\emptyset$ Current T: $\{V(G)\}$

(2) Choose vertex $X\in V(T)$ such that $|X|\geq 2$ as a vertex set of G. If no such X go to (6)



(3) Choose $s,t\in X$ with $s\neq t$:

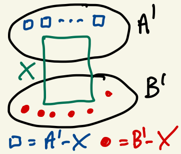
For each connected component C of $T-X$

Contract $S_C=\bigcup_{Y\in V(C)} Y$ into one vertex v_C (in G)

Now the resulting graph G' has $V(G')=X\cup\{v_C \mid C \text{ is a component of } T-X\}$

(4) Find a minimum (s,t)-cut (A',B') in (G',u') and set

$$A=\left(\bigcup_{v_C\in A'-X} S_C\right)\cup(A'\cap X),\quad B=\left(\bigcup_{v_C\in B'-X} S_C\right)\cup(B'\cap X)$$



(5) Let $V(T)=(V(T)-\{X\})\cup\{(A\cap X),(B\cap X)\}$

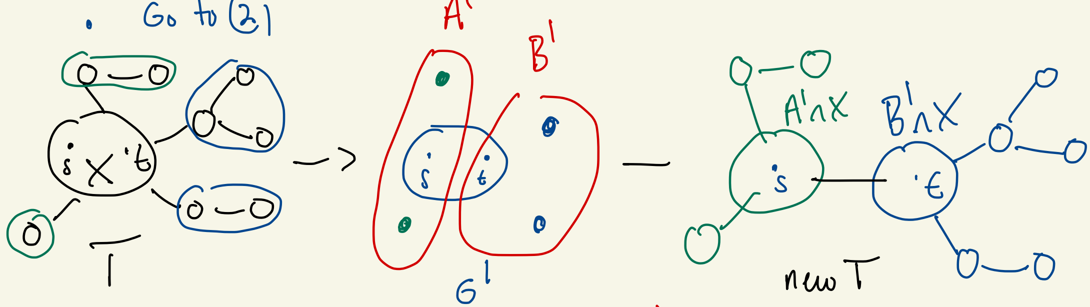
- $\forall e=\{X,Y\}\in E(T)$ s.t. e is incident to X in T do:

  If $Y\in A$ then $e':=\{A\cap X, Y\}$ else $e':=\{B\cap X, Y\}$

  $E(T):=(E(T)-e)+e'$ and $w(e'):=w(e)$

- $E(T):=E(T)+\{A\cap X, B\cap X\}$ with $w(A\cap X, B\cap X)=u'(\delta_{G'}(A'))$

- Go to (2)

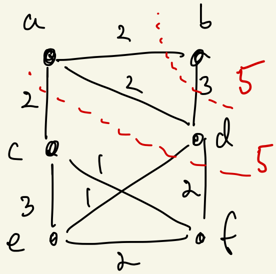


(6) Replace all $\{x\}\in V(T)$ by the vertex x

Replace all $(\{x\},\{y\})\in E(T)$ by $\{x,y\}$

Return T

Example of the algorithm



$T = \bullet$ $\{a,c,d,e,f\},\{b\}$ min (a,b)-cut cap $= 5$

$\downarrow$

X V-X

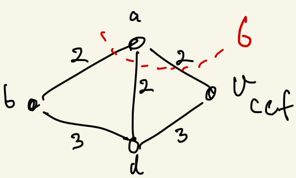
$T = \{a,c,d,e,f\} \overset{5}{—} b$

Choose $X = \{a,c,d,e,f\}$ and the pair (a,c)

$G' = G$ so no contraction. Minimum (a,c) cut has size 5

e.g $\{a,b,d\},\{c,e,f\}$ new $T = (c,e,f) \overset{5}{—} (a,d) \overset{5}{—} (b)$

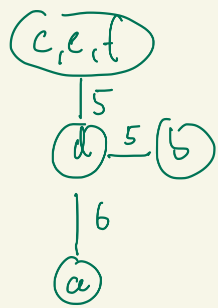
choose $X = \{a,d\}$ and contract $\{c,e,f\}$:



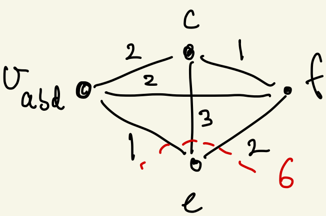
Minimum (a,d)-cut has size 6

$\{a\},\{b,d,v_{cef}\}$ is min cut

new T

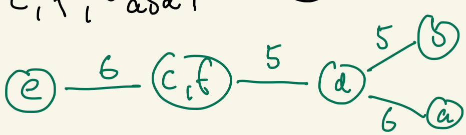


Choose $X = \{c,e,f\}$ and contract $\{a,b,d\}$

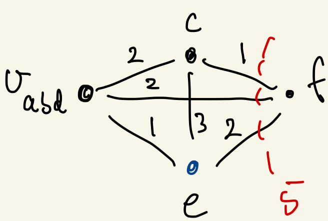


$\{e\},\{c,f,v_{abd}\}$ min (c,e)-cut

new T

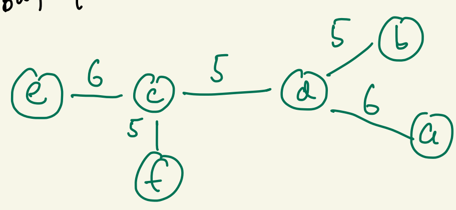


Contract $\{a,b,d\}$



$\{f\},\{v_{abd},c,d\}$ min (c,f)-cut

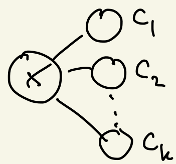
Final T



Correctness of the algorithm:

Lemma 8.33 Each time step (4) ends we have
(a) $A \cup B = V(G)$ and (b) (A, B) is a minimum (s,t)-cut in (G, u)

Proof (a) is clear
(b) clearly holds after the first execution of (4) as $G' = G$
We now show that (b) is preserved in each iteration of (4)



Let $C_1, C_2, \ldots C_k$ be the connected components of $T - X$
Consider contracting $C_1, C_2 \ldots C_k$ one by one:
For $i = 0, 1, 2 \ldots k$ the (G_i, u_i) arise from (G, u) by contracting each of $S_{C_1}, S_{C_2}, \ldots, S_{C_i}$ to a single vertex v_{C_j} for $j \in [i]$
Thus (G_k, u_k) is (G', u') after executing (3) in the algorithm

Claim $\forall$ minimum (s,t)-cut $(A_i, V(G_i) - A_i)$ in (G_i, u_i):
$(A_{i-1}, V(G_{i-1}) - A_{i-1})$ is a minimum (s,t)-cut in (G_{i-1}, u_{i-1})
where $A_{i-1} = \begin{cases} (A_i - v_{C_i}) \cup S_{C_i} & \text{if } v_{C_i} \in A_i \\ A_i & \text{if } v_{C_i} \notin A_i \end{cases}$

Applying the claim in the order $k, k-1, \ldots, 2, 1$ proves (b) proof of the claim:

Let $(A_i, V(G_i)-A_i)$ be a minimum (s,t)-cut in (G_i, u_i)

Since we assumed that (6) holds for the previous iterations (induction)

$(S_{C_i}, V(G)-S_{C_i})$ is a minimum (s_i,t_i)-cut for some $s_i \in S_{C_i}$ and $t_i \notin S_{C_i}$

We have $s,t \in V(G)-S_{C_i}$ so lemma 8.32 implies that

$(A_{i-1}, V(G_{i-1})-A_{i-1})$ is a minimum (s,t)-cut in (G_{i-1}, u_{i-1})

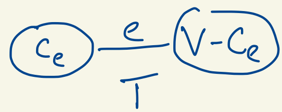
Here (G_{i-1}, u_{i-1}) plays the role of (G,u) in lemma 8.32 and

$s'=s$, $t'=t$.

Lemma 8.34 At any time during the execution of the algorithm

until (6) is reached the following invariant $(\square)+(*)$ holds:

$(\square)$ $\forall e \in E(T)$ $\quad w(e) = u\left(\delta_G\left(\bigcup_{Z\in C_e} Z\right)\right)$ where



$(*)$ $\forall e=\{P,Q\} \in E(T)$ $\exists p \in P, q \in Q$ such that $\lambda_{pq} = w(e)$

Proof The invariant trivially holds when the algorithm starts as $E(T)=\emptyset$

Assume the invariant holds in the first $i-1$ iterations of (2)-(5) and

Consider now a vertex X of T chosen in step (2) in the i'th iteration

and let s,t,A',B',A,B be as in steps (3)+(4)

Only edges incident to X are affected in step (5)

The new edge from $A\cap X$ to $B \cap X$ is set correctly as $s \in A\cap X$, $t \in B\cap X$

and $\lambda_{st} = w(e)$. So look at some edge $e^* = \{X, Y\}$ and assume w.l.o.g

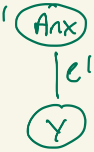
that it became an edge $A\cap X - Y$ (argument for $B\cap X - Y$ analogous)

Since ($\square$), ($*$) hold for e before iteration i we see that ($\square$) still holds for $e' = (A \cap X, Y)$ as $w(e') = w(e) = u\left(\delta_G\left(\bigcup_{z \in C_e} z\right)\right)$

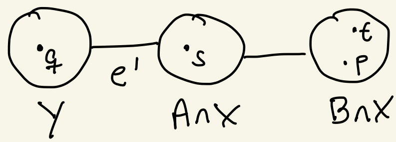
It remains to prove that ($*$) holds for e':

let $p \in X$, $q \in Y$ have $\lambda_{pq} = w(e)$ (they exist as ($*$) holds after iteration $i-1$)

If $p \in A \cap X$ we are done as $q \in Y$ and e moved to e'



So assume $p \in B \cap X$:



If $\lambda_{sq} = \lambda_{pq}$ we are done as then we have

$\lambda_{sq} = \lambda_{pq} = w(e) = w(e')$ ($w(e')$ is set to $w(e)$ in step (5))

So we want to prove $\lambda_{sq} = \lambda_{pq}$

Lemma 8.30 gives: $\lambda_{sq} \geq \min\{\lambda_{st}, \lambda_{tp}, \lambda_{pq}\}$

By lemma 8.33(b) (A, B) is a min (s,t)-cut and $s, q \in A$ so λ_{sq} does not change when we contract B (by lemma 8.32)

We also have $t, p \in B$ so λ_{sq} does not change if we add an edge $t-p$ with ∞ capacity.

This shows that

$$\lambda_{sq} \geq \min\{\lambda_{st}, \lambda_{pq}\} \quad (\Delta)$$

$\lambda_{st} \geq \lambda_{pq}$ since (A,B) is also a (p,q)-cut

$w(e)$ is the capacity of a cut separating X and Y and thus separating s and q (as $s \in X$, $q \in Y$)

This shows that $\lambda_{sq} \leq w(e) = \lambda_{pq}$

so (Δ) implies that $\lambda_{sq} = \lambda_{pq}$ proving that $(*)$ holds for e' after iteration i. $\square$.

Theorem 8.35 The algorithm is correct and it finds a Gomory-Hu tree in polynomial time

Proof: clear that the algorithm is polynomial (running time bounded by time for $(n-1)$ max flow calculations)

Let T be the output of the algorithm. Clearly $V(T) = V$

Let T be the output of the algorithm. Let $s,t \in V$ and let P_{st} be the unique (s,t)-path in T

$\forall e \in E(P_{st})$ let $C_e, V - C_e$ be the connected components of $T-e$

Each of $(C_e, V-C_e)$ is an (s,t)-cut so we have

$$\lambda_{st} \leq \min\{u(\delta(C_e)) \mid e \in P_{s,t}\} \qquad (\nabla)$$

On the other hand, applying Lemma 8.30 repeatedly gives

$$\lambda_{st} \geq \min\{\lambda_{vw} \mid (v,w) \in E(P_{s,t})\} \qquad (*)$$

Apply Lemma 8.34 to the situation just be step (6) is executed

$(*)$ implies that $\lambda_{st} \geq \min\{u(\delta(C_e)) \mid e \in P_{s,t}\}$

Now (∇) implies that $\lambda_{st} = \min\{u(\delta(C_e)) \mid e \in P_{s,t}\}$ $\square$.