# All-pairs maximum flows

## Combinatorial Optimization

Giovanni Righini

University of Milan

UNIVERSITÀ DEGLI STUDI DI MILANO

# All-pairs max flows

A maximum flow for each pair of nodes $s$ and $t$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be obviously found by running any maxflow algorithm $O(n^2)$ times, where $n = |\mathcal{V}|$.

For undirected graphs it is possible to do better: a Gomory-Hu tree, representing all minimum cuts in the graph, can be found in $O(n)$ runs of any maxflow algorithm.

# Gomory-Hu tree (1961)

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a graph and $u : \mathcal{E} \mapsto \Re_+$ a capacity function.

A Gomory-Hu tree for $\mathcal{G}$ and $u$ is a tree $\mathcal{T} = (\mathcal{V}, \mathcal{F})$ (not necessarily a subgraph of $\mathcal{G}$) such that $\forall e = [s, t] \in \mathcal{F}$, $\delta(S)$ is a minimum capacity $(s, t)$-cut of $\mathcal{G}$, where $S$ is one of the two components of $\mathcal{T} \setminus \{e\}$.
+ **Property 1.** For any $\mathcal{G}$ and $u$ a Gomory-Hu tree exists.

**Property 2.** A Gomory-Hu tree can be computed in $O(n)$ maxflow computations.

# An example

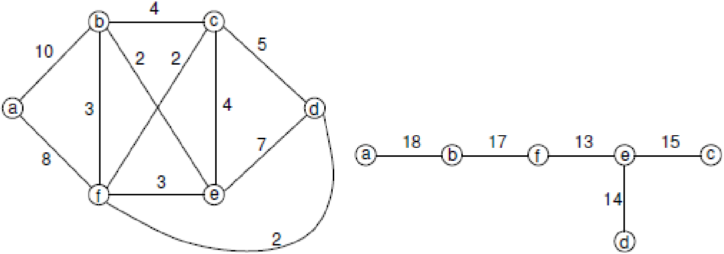
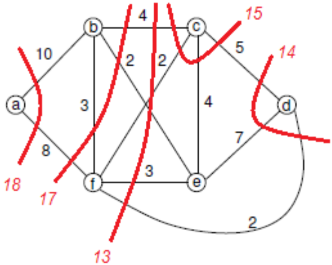
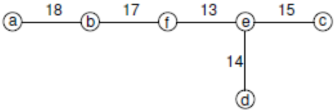



Figure: From Vazirani, *Approximation algorithms*, Springer, 2004.

# An example




| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| a | | 18 | 13 | 13 | 13 | 17 |
| b | | | 13 | 13 | 13 | 17 |
| c | | | | 14 | 15 | 13 |
| d | | | | | 14 | 13 |
| e | | | | | | 13 |
| f | | | | | | |

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| a | | a | abf | abf | abf | ab |
| b | | | abf | abf | abf | ab |
| c | | | | d | c | abf |
| d | | | | | d | abf |
| e | | | | | | abf |
| f | | | | | | |

# Gomory-Hu trees and min cuts

For any two distinct vertices $s$ and $t$ in $\mathcal{V}$, let $r(s,t)$ be the minimum capacity of an $(s,t)$-cut.

**Property.** The following triangle inequality holds:

$$r(i,j) \geq \min_{k \in \mathcal{V} \setminus \{i,j\}} \{r(i,k), r(k,j)\} \;\; \forall i,j \in \mathcal{V} : i \neq j.$$

# Gomory-Hu trees and min cuts

Let $s$ and $t$ be two distinct vertices in graph $\mathcal{G}$ with capacity function $u$.
Let $\mathcal{T}$ be a Gomory-Hu tree for $\mathcal{G}$ and $u$.
Let $\mathcal{P}$ be the (unique) $(s,t)$-path in $\mathcal{T}$.
Let $e = [i,j]$ the edge of $\mathcal{P}$ with minimum value of $r$.
Let $S$ be one of the two components of $\mathcal{T} \setminus \{e\}$.

**Theorem.** The capacity $r(s,t) = r(i,j)$ and $\delta(S)$ is a minimum capacity $(s,t)$-cut.

**Proof.** Inductively, $r(s,t) \geq r(i,j)$.
$\delta(S)$ is an $(s,t)$-cut, because the deletion of $e = [i,j]$ disconnects $s$ from $t$.
$r(s,t) \leq u(\delta(S))$ by definition of $r(s,t)$.
$u(\delta(S)) = r(i,j)$ by the property of Gomory-Hu trees.
Therefore $r(s,t) \leq u(\delta(S)) = r(i,j)$.
Therefore $r(s,t) = r(i,j)$.

# A lemma

To prove the existence of a Gomory-Hu tree for any graph $\mathcal{G}$ and capacity function $u$, a lemma in needed.

**Lemma.**

- Let $s$ and $t$ be two distinct vertices in graph $\mathcal{G}$ with capacity function $u$,
- let $\delta(S)$ be a minimum capacity $(s, t)$-cut in $\mathcal{G}$,
- let $i$ and $j$ be two distinct vertices in $S$.

Then, there exists a minimum capacity $(i, j)$-cut $\delta(X)$ with $X \subseteq S$.

# Existence: proof of the lemma

**Proof.** Consider a minimum $(i,j)$-cut $\delta(X)$.
W.l.o.g. assume $s \in S$ and $t \notin S$ (otherwise swap $s$ and $t$).
W.l.o.g. assume $s \in X$ (otherwise swap $X$ and $S \setminus X$).
W.l.o.g. assume $i \in X$ and $j \notin X$ (otherwise swap $i$ and $j$).
Two cases may occur: $t \notin X$ and $t \in X$.

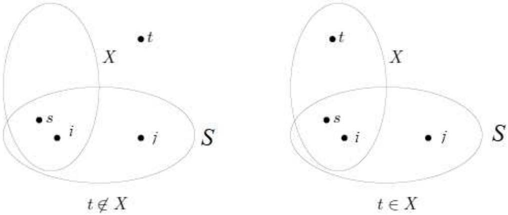


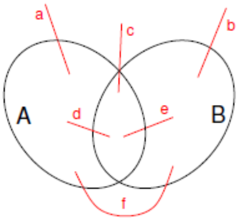

Figure: From by A. Schrijver, *Combinatorial optimization*, Springer 2003, page 249.

# Two inequalities

For any two vertex sets $A$ and $B$

$$u(\delta(A \cap B)) + u(\delta(A \cup B)) \leq u(\delta(A)) + u(\delta(B)). \qquad (1)$$

$$u(\delta(A \backslash B)) + u(\delta(B \backslash A)) \leq u(\delta(A)) + u(\delta(B)). \qquad (2)$$




(1): $(c + d + e) + (a + b + c) \leq (a + c + e + f) + (b + c + d + f)$.
(2): $(a + d + f) + (b + e + f) \leq (a + c + e + f) + (b + c + d + f)$.

# Proof of the lemma - case 1

Both $\delta(S \cap X)$ and $\delta(S \setminus X)$ are $(i, j)$-cuts.

If $t \notin X$, then $\delta(S \cup X)$ is an $(s, t)$-cut.

For inequality (1):

$$u(\delta(S \cap X)) + u(\delta(S \cup X)) \leq u(\delta(S)) + u(\delta(X)).$$

Since $\delta(S \cup X)$ an $(s, t)$-cut and by definition $\delta(S)$ is a minimum capacity $(s, t)$-cut,

$$u(\delta(S \cup X)) \geq u(\delta(S)).$$

Hence

$$u(\delta(S \cap X)) \leq u(\delta(X)).$$

Since by definition $\delta(X)$ is a minimum capacity $(i, j)$-cut, then also $\delta(S \cap X)$ is a minimum capacity $(i, j)$-cut.

# Proof of the lemma - case 2

If $t \in X$, then $\delta(X \setminus S)$ is an $(s,t)$-cut.

For inequality (2):

$$u(\delta(S \setminus X)) + u(\delta(X \setminus S)) \leq u(\delta(S)) + u(\delta(X)).$$

Since $\delta(X \setminus S)$ is an $(s,t)$-cut and by definition $S$ is a minimum capacity $(s,t)$-cut,

$$u(\delta(X \setminus S)) \geq u(\delta(S)).$$

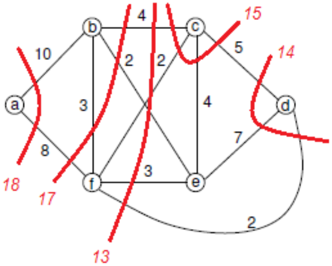
Hence

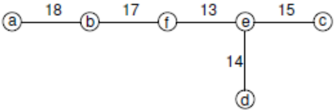
$$u(\delta(S \setminus X)) \leq u(\delta(X)).$$

Since by definition $X$ is a minimum capacity $(i,j)$-cut, then also $\delta(S \setminus X)$ is a minimum capacity $(i,j)$-cut.

# The lemma

A different (more intuitive) way to state the lemma is:
*minimum capacity cuts cannot intersect.*



# Existence of the Gomory-Hu tree

**Theorem.** For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and any capacity function $u : \mathcal{E} \mapsto \Re_+$, there exists a Gomory-Hu tree for $\mathcal{G}$ and $u$.

**Proof.** The proof is by induction.

For each $R \subseteq \mathcal{V}$ consider

- a tree $\mathcal{T} = (R, \mathcal{F})$,
- a partition of $\mathcal{V}$ into subsets $C_r \ \forall r \in R$, such that:
  - $r \in C_r \ \ \forall r \in R$;
  - for each edge $e = [s, t]$ in $\mathcal{F}$, $\delta(S)$ is a minimum capacity $(s, t)$-cut, where $S = \bigcup_{k \in K} C_k$ and $K$ is a component of $T \setminus \{e\}$.

$C_r$ is made by the vertices that lie on the same side as $r$ in some minimum capacity cut.

# Existence of the Gomory-Hu tree

If $|R| = 1$, the conditions are trivially satisfied. Then, assume $|R| \geq 2$.

Let $\delta(W)$ be a minimum capacity cut separating at least one pair of vertices in $R$.

Contract $\mathcal{V} \setminus W$ into a single vertex, $v'$, yielding graph $\mathcal{G}'$.
Consider the vertex subset $R' = R \cap W$.
By induction, $\mathcal{G}'$ has a Gomory-Hu tree $(R', \mathcal{F}')$, with a partition of $R'$ in subsets $C'_r \; \forall r \in R'$.

Similarly, contract $W$ into a single vertex, $v''$, yielding graph $\mathcal{G}''$.
Consider the vertex subset $R'' = R \setminus W$.
By induction, $\mathcal{G}''$ has a Gomory-Hu tree $(R'', \mathcal{F}'')$, with a partition of $R''$ in subsets $C''_r \; \forall r \in R''$.

# Existence of the Gomory-Hu tree

Let $r' \in R'$ be such that $v' \in C'_{r'}$.
Let $r'' \in R''$ be such that $v'' \in C''_{r''}$.

Consider $\mathcal{F} = \mathcal{F}' \cup \mathcal{F}'' \cup [r', r'']$.
Consider $C_{r'} = C'_{r'} \setminus \{v'\}$ and $C_r = C'_r$ for all the other $r \in R'$.
Consider $C_{r''} = C''_{r''} \setminus \{v''\}$ and $C_r = C''_r$ for all the other $r \in R''$.

Now the tree $\mathcal{T} = (R, \mathcal{F})$ and the partition $C_r$ form a Gomory-Hu tree for $R$.

**Proof.**
For any $e \in \mathcal{F}$ with $e \neq [r', r'']$, the Gomory-Hu properties follow from the Lemma.

If $e = [r', r'']$, then $S = W$ and $r(W)$ is a minimum capacity $(r', r'')$-cut, because it is one with minimum capacity over all cuts separating at least one pair of vertices in $R$.

Procedure $GomoryHuTree(\mathcal{G}, R)$

**if** $|R| = 1$ **then**

// Recursion base //

Select $r$ such that $R = \{r\}$

$\mathcal{T} \leftarrow (\{r\}, \emptyset)$

$C_r \leftarrow \mathcal{V}$; $C \leftarrow \{C_r\}$

**else**

Select $r_1, r_2 \in R$

$\delta(W) \leftarrow MinCut(r_1, r_2)$

// Create two sub-instances of the problem //

$\mathcal{G}_1 \leftarrow Shrink(\mathcal{G}, \mathcal{V} \backslash W, v_1)$; $R_1 \leftarrow R \cap W$

$\mathcal{G}_2 \leftarrow Shrink(\mathcal{G}, W, v_2)$; $R_2 \leftarrow R \backslash W$

// Recursive step //

$(\mathcal{T}_1, C^1) \leftarrow GomoryHuTree(\mathcal{G}_1, R_1)$

$(\mathcal{T}_2, C^2) \leftarrow GomoryHuTree(\mathcal{G}_2, R_2)$

// Vertex selection //

Select $r' \in R_1$ such that $v_1 \in C^1_{r'}$

Select $r'' \in R_2$ such that $v_2 \in C^2_{r''}$

// Compute tree and partitions for $(\mathcal{G}, R)$ //

$\mathcal{T} \leftarrow (R_1 \cup R_2, \mathcal{E}(T_1) \cup \mathcal{E}(T_2) \cup \{[r', r'']\})$

$C \leftarrow ComputePartitions(R_1, R_2, C^1, C^2, r', r'')$

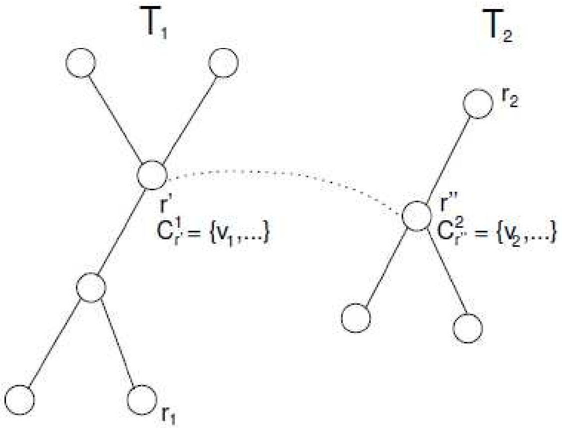
**return** $(\mathcal{T}, C)$

# Computing a Gomory-Hu tree

Procedure $ComputePartitions(R_1, R_2, C^1, C^2, r', r'')$
// Remove $v_1$ and $v_2$ from $C_{r'}$ and $C_{r''}$ //
**for** $r \in R_1 : r \neq r'$ **do**
    $C_r \leftarrow C^1_r$
$C_{r'} \leftarrow C^1_{r'} \setminus \{v_1\}$
**for** $r \in R_2 : r \neq r''$ **do**
    $C_r \leftarrow C^2_r$
$C_{r''} \leftarrow C^2_{r''} \setminus \{v_2\}$
**return** $C$

Vertices in $C_r$ must appear on the same side of $r$ in some min cut.

# Computing a Gomory-Hu tree




From two Gomory-Hu trees a larger one is obtained, recursively.

# Computing a Gomory-Hu tree

When two trees are merged in a larger one,

- all the edges of $\mathcal{T}_1$ and $\mathcal{T}_2$ correspond to min cuts, because $\mathcal{T}_1$ and $\mathcal{T}_2$ are Gomory-Hu trees and because the Lemma guarantees that each subtree can be considered separately because min cuts do not cross;
- the new edge $[r', r'']$ also corresponds to a min cut (this can be easily proven by contradiction).

Vertices $r_1$ and $r_2$ can be selected arbitrarily: the correctness of the algorithm is not affected.

The number of calls to *MinCut* is the same as the number of edges in the final Gomory-Hu tree, that is $n - 1$.