CS 598CSC: Combinatorial Optimization	Lecture date: 2/4/2010
Instructor: Chandra Chekuri	Scribe: David Morrison

Gomory-Hu Trees

(The work in this section closely follows [3])

Let $G = (V, E)$ be an undirected graph with non-negative edge capacities defined by $c : E \to \mathbb{R}$. We would like to be able to compute the *global* minimum cut on the graph (i.e., the minimum over all min-cuts between pairs of vertices s and t). Clearly, this can be done by computing the minimum cut for all $\binom{n}{2}$ pairs of vertices, but this can take a lot of time. Gomory and Hu showed that the number of distinct cuts in the graph is at most $n-1$, and furthermore that there is an efficient tree structure that can be maintained to compute this set of distinct cuts [1] (note that there is also a very nice randomized algorithm due to Karger and Stein that can compute the global minimum cut in near-linear time with high probability [2]).

An important note is that Gomory-Hu trees work because the cut function is both submodular and symmetric. We will see later that *any* submodular, symmetric function will induce a Gomory-Hu tree.

Definition 1. *Given a graph $G = (V, E)$, we define $\alpha_G(u, v)$ to be the value of a minimum u, v cut in G. Furthermore, for some set of vertices U, we define $\delta(U)$ to be the set of edges with one endpoint in U.*

Definition 2. *Let G, c, and α_G be defined as above. Then, a tree $T = (V(G), E_T)$ is a* **Gomory-Hu tree** *if for all $st \in E_T$, $\delta(W)$ is a minimum s, t cut in G, where W is one component of $T - st$.*

The natural question is whether such a tree even exists; we will return to this question shortly. However, if we are given such a tree for an arbitrary graph G, we know that this tree obeys some very nice properties. In particular, we can label the edges of the tree with the values of the minimum cuts, as the following theorem shows (an example of this can be seen in figure 1):

Theorem 1. *Let T be a Gomory-Hu tree for a graph $G = (V, E)$. Then, for all $u, v \in V$, let st be the edge on the unique path in T from u to v such that $\alpha_G(s, t)$ is minimized. Then,*

$$\alpha_G(u, v) = \alpha_G(s, t)$$

and the cut $\delta(W)$ induced by $T - st$ is a u, v minimum cut in G. Thus $\alpha_G(s, t) = \alpha_T(s, t)$ for each $s, t \in V$ where the capacity of an edge st in T is equal to $\alpha_G(s, t)$.

Proof. We first note that α_G obeys a triangle inequality. That is, $\alpha_G(a, b) \geq \min(\alpha_G(a, c), \alpha_G(b, c))$ for any undirected graph G and vertices a, b, c (to see this, note that c has to be on one side or the other of any a, b cut).

Consider the path from u to v in T. We note that if $uv = st$, then $\alpha_G(u, v) = \alpha_G(s, t)$. Otherwise, let $w \neq v$ be the neighbor of u on the u-v path in T. By the triangle inequality mentioned above, $\alpha_G(u, v) \geq \min(\alpha_G(u, w), \alpha_G(w, v))$. If $uw = st$, then $\alpha_G(u, v) \geq \alpha_G(s, t)$; otherwise, by induction on the path length, we have that $\alpha_G(u, v) \geq \alpha_G(w, v) \geq \alpha_G(s, t)$.

However, by the definition of Gomory-Hu trees, we have that $\alpha_G(u, v) \leq \alpha_G(s, t)$, since the cut induced by $T - st$ is a valid cut for u, v. Thus, we have $\alpha_G(u, v) = \alpha_G(s, t)$ and the cut induced by $T - st$ is a u, v minimum cut in G. ∎

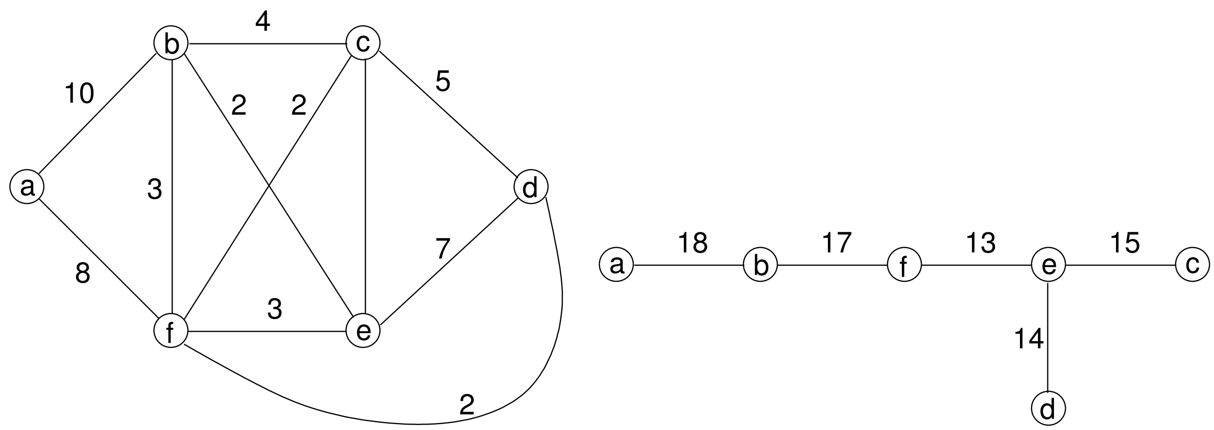

Figure 1: A graph G with its corresponding Gomory-Hu tree [4].

Remark 2. *Gomory-Hu trees can be (and are often) defined by asking for the property described in Theorem 1. However, the proof shows that the basic requirement in Definition 2 implies the other property.*

The above theorem shows that we can represent compactly all of the minimum cuts in an undirected graph. Several non-trivial facts about undirected graphs fall out of the definition and the above result. The only remaining question is "Does such a tree exist? And if so, how does one compute it efficiently?" We will answer both questions by giving a constructive proof of Gomory-Hu trees for any undirected graph G. However, first we must discuss some properties of submodular functions.

Definition 3. *Given a finite set E, $f : 2^E \to \mathbb{R}$ is* **submodular** *if for all $A, B \in 2^E$, $f(A)+f(B) \geq f(A \cup B) + f(A \cap B)$.*

An alternate definition based on the idea of "decreasing marginal value" is the following:

Definition 4. *Given E and f as above, f is submodular if $f(A+e) - f(A) \geq f(B+e) - f(B)$ for all $A \subseteq B$ and $e \in E$.*

To see the equivalence of these definitions, let $f_A(e) = f(A+e) - f(A)$, and similarly for $f_B(e)$. Take any $A, B \subseteq E$ and $e \in E$ such that $A \subseteq B$, and let f be submodular according to definition 3. Then $f(A+e) + f(B) \geq f((A+e) \cup B) + f((A+e) \cap B) = f(B+e) + f(A)$. Rearranging shows that $f_A(e) \geq f_B(e)$. Showing that definition 4 implies definition 3 is slightly more complicated, but can be done (**Exercise**).

There are three types of submodular functions that will be of interest:

1. Arbitrary submodular functions
2. Non-negative (range is $[0, \infty)$). Two subclasses of non-negative submodular functions are monotone ($f(A) \leq f(B)$ whenever $A \subseteq B$) and non-monotone.
3. Symmetric submodular functions where $f(A) = f(E \setminus A)$ for all $A \subseteq E$.

As an example of a submodular function, consider a graph $G = (V, E)$ with capacity function $c : E \to \mathbb{R}^+$. Then $f : 2^V to \mathbb{R}^+$ defined by $f(A) = c(\delta(A))$ (i.e., the capacity of a cut induced by a set A) is submodular.

To see this, notice that $f(A)+f(B)=a+b+2c+d+e+2f$, for any arbitrary A and B, and a,b,c,d,e,f are as shown in figure 2. Here, a (for example) represents the total capacity of edges with one endpoint in A and the other in $V \setminus (A \cup B)$. Also notice that $f(A \cup B)+f(A \cap B) = a+b+2c+d+e$, and since all values are positive, we see that $f(A)+f(B) \geq f(A \cup B)+f(A \cap B)$, satisfying definition 3.

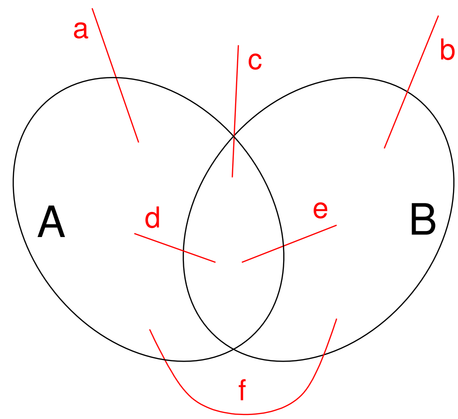

Figure 2: Given a graph G and two sets $A, B \subseteq V$, this diagram shows all of the possible classes of edges of interest in G. In particular, there could be edges with both endpoints in $V \setminus (A \cup B)$, A, or B that are not shown here.

Exercise 1. *Show that cut function on the vertices of a directed graph is submodular.*

Another nice property about this function f is that it is **posi-modular**, meaning that $f(A)+f(B) \geq f(A-B)+f(B-A)$. In fact, posi-modularity follows for any symmetric submodular function:

$$\begin{aligned} f(A)+f(B)=f(V-A)+f(B) &\geq f((V-A)\cap B)+f((V-A)\cup B) \\ &= f(B-A)+f(V-(A-B)) \\ &= f(B-A)+f(A-B) \end{aligned}$$

We use symmetry in the first and last lines above. In fact, it turns out that the above two properties of the cut function are the *only* two properties necessary for the proof of existence of Gomory-Hu trees. As mentioned before, this will give us a Gomory-Hu tree for *any* non-negative symmetric submodular function. We now prove the following lemma, which will be instrumental in constructing Gomory-Hu trees:

Key Lemma. *Let $\delta(W)$ be an s,t minimum cut in a graph G with respect to a capacity function c. Then for any $u,v \in W, u \neq v$, there is a u,v minimum cut $\delta(X)$ where $X \subseteq W$.*

Proof. Let $\delta(X)$ be any u,v minimum cut that crosses W. Suppose without loss of generality that $s \in W, s \in X$, and $u \in X$. If one of these are not the case, we can invert the roles of s and t or X and $V \setminus X$. Then there are two cases to consider:

Case 1: $t \notin X$ (see figure 3). Then, since c is submodular,

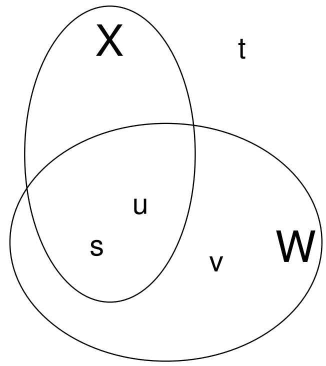

Figure 3: $\delta(W)$ is a minimum s,t cut. $\delta(X)$ is a minimum u,v cut that crosses W. This diagram shows the situation in Case 1; a similar picture can be drawn for Case 2

$$c(\delta(X)) + c(\delta(W)) \geq c(\delta(X \cap W)) + c(\delta(X \cup W)) \tag{1}$$

But notice that $\delta(X \cap W)$ is a u,v cut, so since $\delta(X)$ is a minimum cut, we have $c(\delta(X \cap W)) \geq c(\delta(X))$. Also, $X \cup W$ is a s,t cut, so $c(\delta(X \cup W)) \geq c(\delta(W))$. Thus, equality holds in equation (1), and $X \cap W$ is a minimum u,v cut.

Case 2: $t \in X$. Since c is posi-modular, we have that

$$c(\delta(X)) + c(\delta(W)) \geq c(\delta(W \setminus X)) + c(\delta(X \setminus W)) \tag{2}$$

However, $\delta(W \setminus X)$ is a u,v cut, so $c(\delta(W \setminus X)) \geq c(\delta(X))$. Similarly, $\delta(X \setminus W)$ is an s,t cut, so $c(\delta(X \setminus W)) \geq c(\delta(W))$. Therefore, equality holds in equation (2), and $W \setminus X$ is a u,v minimum cut.

■

The above argument shows that minimum cuts can be **uncrossed**, a technique that is useful in many settings. In order to construct a Gomory-Hu tree for a graph, we need to consider a slightly generalized definition:

Definition 5. *Let $G = (V, E), R \subseteq V$. Then a* **Gomory-Hu tree for R in G** *is a pair consisting of $T = (R, E_T)$ and a partition $(C_r \mid r \in R)$ of V associated with each $r \in R$ such that*

1. *For all $r \in R$, $r \in C_r$*

2. *For all $st \in E_T$, $T - st$ induces a minimum cut in G between s and t defined by*

$$\delta(U) = \bigcup_{r \in X} C_r$$

where X is the vertex set of a component of $T - st$.

Notice that a Gomory-Hu tree for G is simply a generalized Gomory-Hu tree with $R = V$.

Algorithm 1 GomoryHuAlg(G, R)

```
if |R| = 1 then
  return T = ({r}, ∅), C_r = V
else
  Let r_1, r_2 ∈ R , and let δ(W) be an r_1, r_2 minimum cut

  ⟪Create two subinstances of the problem⟫
  G_1 = G with V \ W shrunk to a single vertex, v_1; R_1 = R ∩ W
  G_2 = G with W shrunk to a single vertex, v_2; R_2 = R \ W

  ⟪Now we recurse⟫
  T_1, (C_r^1 | r ∈ R_1) = GomoryHuAlg(G_1, R_1)
  T_2, (C_r^2 | r ∈ R_2) = GomoryHuAlg(G_2, R_2)

  ⟪Note that r′, r″ are not necessarily r_1, r_2!⟫
  Let r′ be the vertex such that v_1 ∈ C_{r′}^1
  Let r″ be the vertex such that v_2 ∈ C_{r″}^2

  ⟪See figure 4⟫
  T = (R_1 ∪ R_2, E_{T_1} ∪ E_{T_2} ∪ {rr′})
  (C_r | r ∈ R) = ComputePartitions(R_1, R_2, C_r^1, C_r^2, r′, r″)
  return T, C_r
end if
```

Algorithm 2 ComputePartitions($R_1, R_2, C_r^1, c_r^2, r', r''$)

```
⟪We use the returned partitions, except we remove v_1 and v_2 from C_{r′} and C_{r″}, respectively⟫
For r ∈ R_1, r ≠ r′, C_r = C_r^1
For r ∈ R_1, r ≠ r″, C_r = C_r^2
C_{r′} = C_{r′}^1 − {v_1}, C_{r″} = C_{r″}^2 − {v_2}
return (C_r | r ∈ R)
```

Intuitively, we associate with each vertex v in the tree a "bucket" that contains all of the vertices that have to appear on the same side as v in some minimum cut. This allows us to define the algorithm GomoryHuAlg.

Theorem 3. GomoryHuAlg *returns a valid Gomory-Hu tree for a set R.*

Proof. We need to show that any $st \in E_T$ satisfies the "key property" of Gomory-Hu trees. That is, we need to show that $T - st$ induces a minimum cut in G between s and t. The base case is trivial. Then, suppose that $st \in T_1$ or $st \in T_2$. By the Key Lemma, we can ignore all of the vertices outside of T_1 or T_2, because they have no effect on the minimum cut, and by our induction hypothesis, we know that T_1 and T_2 are correct.

Thus, the only edge we need to care about is the edge we added from r' to r''. First, consider the simple case when $\alpha_G(r_1, r_2)$ is minimum over all pairs of vertices in R. In this case, we see that in particular, $\alpha_G(r_1, r_2) \leq \alpha_G(r', r'')$, so we are done.

However, in general this may not always be the case. Let $\delta(W)$ be a minimum cut between r_1 and r_2, and suppose that there is a smaller r', r'' minimum cut $\delta(X)$ than what W induces; that is $c(\delta(X)) < (\delta(W))$. Assume without loss of generality that $r_1, r' \in W$. Notice that if $r_1 \in X$, we

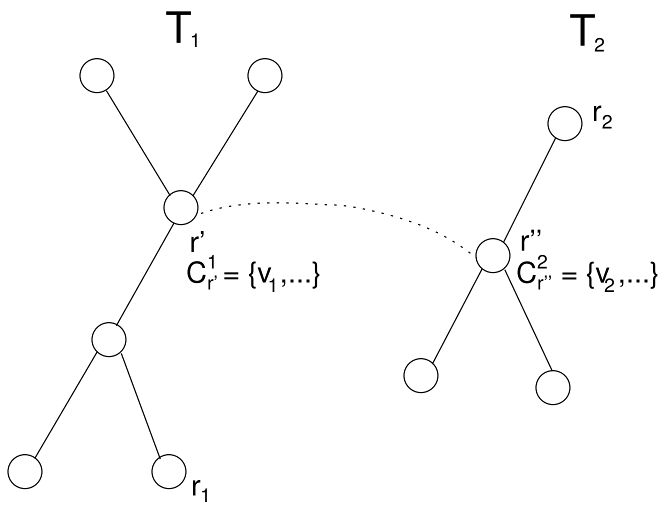

Figure 4: T_1 and T_2 have been recursively computed by GomoryHuAlg. Then we find r' and r'' such that v_1 (the shrunken vertex corresponding to $V \setminus W$ in T_1) is in the partition of r', and similarly for r'' and v_2. Then, to compute T, we connect r' and r'', and recompute the partitions for the whole tree according to ComputePartitions.

have a smaller r_1, r_2 cut than $\delta(W)$, and similarly if $r_2 \in X$. So, it is clear that X separates r_1 and r'. By our key lemma, we can then uncross and assume that $X \subseteq W$.

Now, consider the path from r' to r_1 in T_1. There exists an edge uv on this path such that the weight of uv in T_1, $w_1(uv)$, is at most $c(\delta(X))$. Because T_1 is a Gomory-Hu tree, uv induces an r_1, r_2 cut in G of capacity $w_1(uv)$ (since $v_1 \in C^1_{r'}$). But this contradicts the fact that W is a r_1, r_2 minimum cut. Threfore, e can pick r_1 and r_2 arbitrarily from R_1 and R_2, and GomoryHuAlg is correct. ∎

This immediately implies the following corollary:

Corollary 4. *A Gomory-Hu tree for $R \subseteq V$ in G can be computed in the time needed to compute $|R| - 1$ minimum-cuts in graphs of size at most that of G.*

Finally, we present the following alternative proof of the last step of theorem 3 (that is, showing that we can choose r_1 and r_2 arbitrarily in GomoryHuAlg). As before, let $\delta(W)$ be an r_1, r_2 minimum cut, and assume that $r_1 \in W, r_2 \in V \setminus W$. Assume for simplicity that $r_1 \neq r'$ and $r_2 \neq r''$ (the other cases are similar). We claim that $\alpha_{G_1}(r_1, r') = \alpha_G(r_1, r') \geq \alpha_G(r_1, r_2)$. To see this, note that if $\alpha_{G_1}(r_1, r') < \alpha_G(r_1, r_2)$, there is an edge $uv \in E_{T_1}$ on the path from r_1 to r' that has weight less than $\alpha_G(r_1, r_2)$, which gives a smaller r_1, r_2 cut in G than W (since $v_1 \in C^1_{r'}$). For similar reasons, we see that $\alpha_G(r_2, r'') \geq \alpha_G(r_1, r_2)$.

Thus, by the triangle inequality we have

$$\alpha_G(r', r'') \geq \min(\alpha_G(r', r_1), \alpha_G(r'', r_2), \alpha_G(r_1, r_2) \geq \alpha_G(r_1, r_2)$$

which completes the proof.

Gomory-Hu trees allow one to easily show some facts that are otherwise hard to prove directly. Some examples are the following.

Exercise 2. *For any undirected graph there is a pair of nodes s, t and an s-t minimum cut consisting of a singleton node (either s or t). Such a pair is called a* pendant *pair.*

Exercise 3. *Let G be a graph such that $deg(v) \geq k$ for all $v \in V$. Show that there is some pair s, t such that $\alpha_G(s,t) \geq k$.*

Notice that the proof of the correctness of the algorithm relied only on the key lemma which in turn used only the symmetry and submodularity of the cut function. One can directly extend the proof to show the following theorem.

Theorem 5. *Let V be a ground set, and let $f : 2^V \to \mathbb{R}^+$ be a symmetric submodular function. Given s, t in V, define the minimum cut between s and t as*

$$\alpha_f(s,t) = \min_{W \subseteq V, |W \cap \{s,t\}|=1} f(W)$$

Then, there is a Gomory-Hu tree that represents α_f. That is, there is a tree $T = (V, E_T)$ and a capacity function $c : E_T \to \mathbb{R}^+$ such that $\alpha_f(s,t) = \alpha_T(s,t)$ for all s, t in V, and moreover, the minimum cut in T induces a minimum cut according to f for each s, t.

Exercise 4. *Let $G = (V, \xi)$ be a hypergraph. That is, each hyper-edge $S \in \xi$ is a subset of V. Define $f : 2^V \to \mathbb{R}^+$ as $f(W) = |\delta(W)|$, where $S \in \xi$ is in $\delta(W)$ iff $S \cap W$ and $S \setminus W$ are non-empty. Show that f is a symmetric, submodular function.*

References

[1] R. E. Gomory, T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, 1961.

[2] D. R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, vol. 47, 2000.

[3] A. Schrijver. Combinatorial Optimization. Springer-Verlag Berlin Heidelberg, 2003. Chapter 15.4.

[4] V. Vazirani. Approximation Algorithms. Springer, 2004.