Multiway Cut Problem

Given undirected graph $G = (V, E)$
Edge weights $w: E \to \mathcal{R}^+$

Terminals: $T = \{t_1, t_2, \ldots, t_k\} \subseteq V$

Goal: find minimum weight set of edges E' such removing E' separates all terminals

That is, no connected component of G(V,E-E') has two terminals from T

Facts

- $k = 2$, standard s-t cut problem, polynomial time solvable

- Multiway Cut is NP-hard and also APX-hard even for $k = 3$

- Multiway Cut can be solved exactly for fixed k in *planar graphs*

Isolating Cuts

For $t_i \in T$, $E' \subset E$ is an isolating cut if removing edges in E' separates t_i from all other terminals

Minimum weight isolating cut can be computed in polynomial time (How?)

Let $C_1, C_2, \ldots, C_k$ be min weight isolating cuts for $t_1, t_2, \ldots, t_k$

Isolating Cut Heuristic

Let $C_1, C_2, \ldots, C_k$ be min weight isolating cuts for $t_1, t_2, \ldots, t_k$

Assume wlog that $w(C_1) \leq w(C_2) \leq \ldots \leq w(C_k)$

Output $C = C_1 \cup C_2 \cup \ldots \cup C_{k-1}$

Claim: C is a feasible multiway cut

Analysis

Theorem: $w(C) \leq 2(1-1/k)$ OPT

Consider some optimum cut A

Let $G[V_i]$ be the connected component in $G(V,E-A)$ that contains terminal t_i

$A_i = \delta(V_i)$: edges with exactly one end point in V_i

Analysis

Consider some optimum cut A

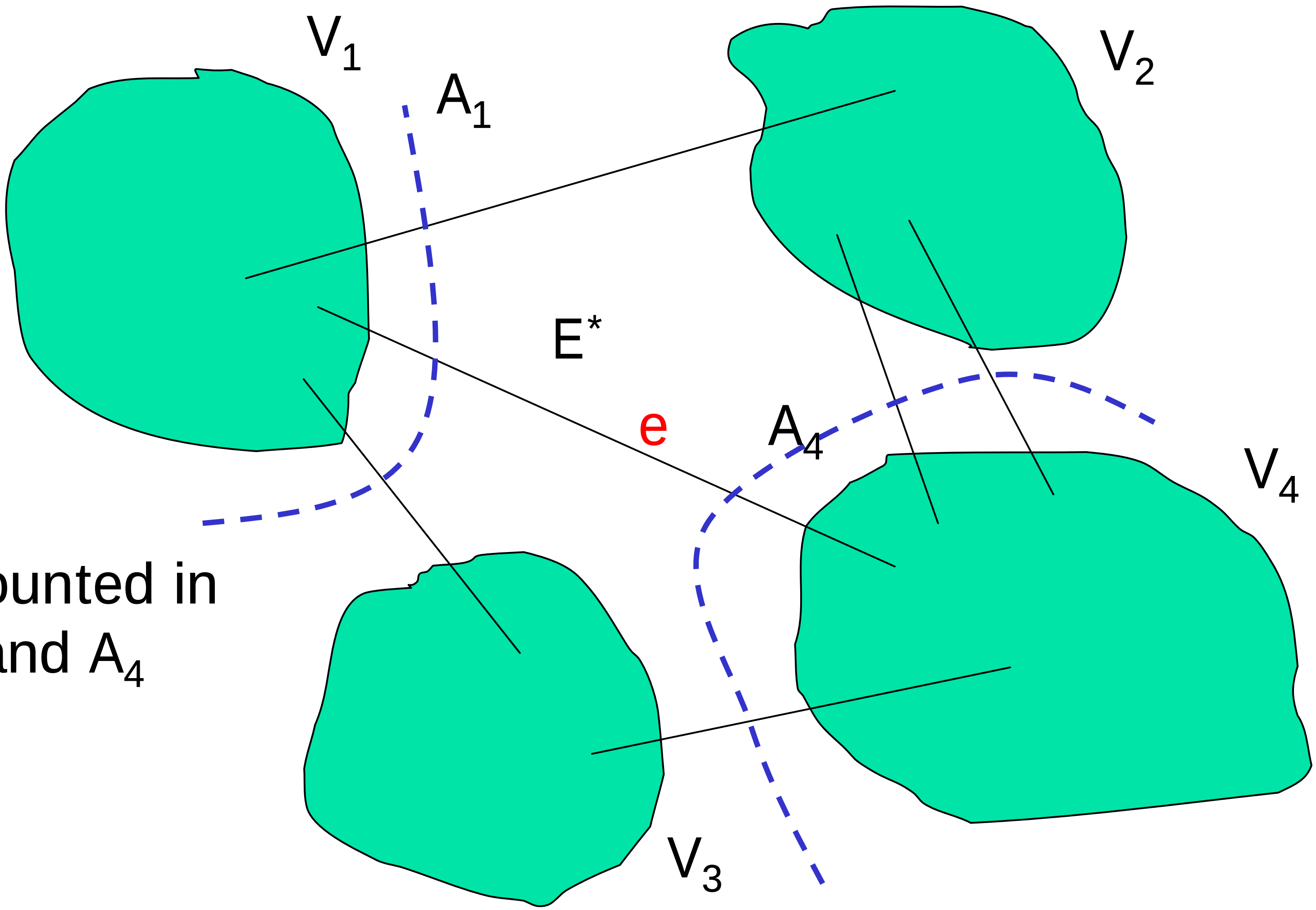
Let $G[V_i]$ be the connected component in $G(V,E-A)$ that contains terminal t_i

$A_i = \delta(V_i)$: edges with exactly one end point in V_i

Claim: $2w(A) = \sum_{i=1}^{k} w(A_i)$

Each edge in A is counted twice

A and A_1, A_2, …, A_k



e counted in A_1 and A_4

Analysis

Claim: $2OPT = \sum_{i=1}^{k} w(A_i)$

A_i is an isolating cut for t_i therefore
for $1 \leq i \leq k$

$w(A_i) \geq w(C_i)$ since C_i was min-wt isolating cut for t_i

implies

$\sum_{i=1}^{k} w(A_i) \geq \sum_{i=1}^{k} w(C_i)$

Analysis

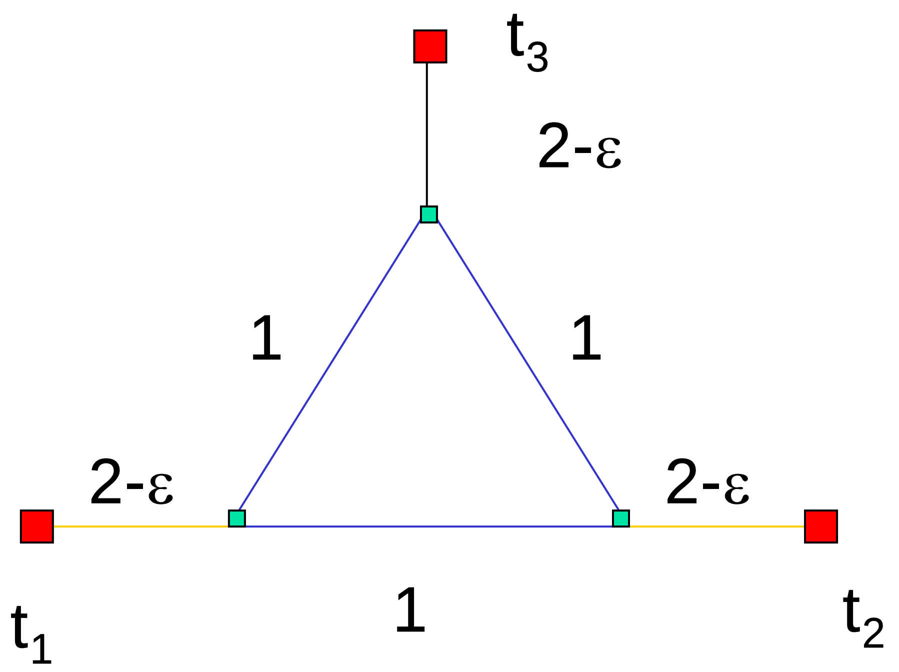
$w(C) = w(C_1) + \ldots w(C_{k-1})$

$\leq (1-1/k) (w(C_1) + \ldots w(C_k))$ (since C_k is the heaviest cut)

$\leq (1-1/k) (w(A_1) + \ldots w(A_k))$

$\leq 2(1-1/k) w(A)$

A Tight Example



Blue: opt cut of value 3

Yellow: algorithm's cut of value 4-2ε

Example can be generalized to large k to reach ratio of (2-2/k)

A Greedy Splitting Algorithm

Start with G

Split into two components such that each contains a terminal

Split *one* of the two components such that each of the three components has a terminal

Split *one* of the three components such that each of the four components has a terminal

… till k components each with terminal

At each step choose a cheapest cut among components

Greedy Splitting Algorithm

Theorem: Greedy splitting also a 2-2/k approximation algorithm for multiway cut

Proof: Exercise

The k-Cut Problem

Given undirected $G=(V,E)$
w: $E \rightarrow \mathcal{R}^+$

integer k

Goal: find minimum weight set of edges to remove such that G is partitioned into k connected components

Facts

- $k=2$ is the global mincut problem, can be solved in polynomial time (near linear time using randomization)
- Can be solved in $O(n^{k^2})$ time – hence polynomial time solvable for fixed k
- NP-hard for arbitrary k

A Greedy Splitting Algorithm

Start with G

Split into two components

Split into three components by splitting one of the two components

…

Choose cheapest split at each stage

Greedy Splitting

Theorem: Greedy splitting is a 2-2/k approximation

Proof is complicated

We do an alternate proof using Gomory-Hu trees

Cut Structure of Undirected Graphs

Given undirected graph $G = (V, E)$
$w: E \rightarrow \mathcal{R}^+$

Let $mc(ab)$ denote weight of min a-b cut in G

There are $n(n-1)/2$ pairs of vertices so potentially $n(n-1)/2$ different min-cut values

However …

Cut Structure of Undirected Graphs

There are n(n-1)/2 pairs of vertices so potentially n(n-1)/2 different min-cut values

However only n-1 distinct cut-values

Moreover magical Gomory-Hu tree!

Gomory-Hu tree for G

$G=(V, E)$ with edge weights w

Gomory-Hu tree $T = (V, E_T)$
$u: E_T \to \mathcal{R}^+$

same vertex set as G

u: weights on edges of T

For each pair (a, b) of vertices of G, their min-cut value mc(ab) in G is *equal to* min cut value in T!

Gomory-Hu tree for G

For each pair (a, b) of vertices of G, their min-cut value mc(ab) in G is *equal to* min cut value in T!

Min-cut in T is min-weight edge in *unique* path connecting a and b in T

In particular for an edge ab in E_T we have $u(ab) = mc(ab)$

Gomory-Hu Tree for G

Can be computed using $O(n)$ s-t cut computations

With each edge ab in E_T we can also associate a min a-b cut C_{ab} of value $u(ab)$

Removing edges in C_{ab} disconnects a from b

k-Cut alg using Gomory-Hu trees

Run Greedy Splitting on T instead of G

Equivalent to picking the k-1 lightest edges in T

Let $e_1, e_2, \ldots, e_{k-1}$ be the chosen edges (from T)

Output $C = C_{e_1} \cup C_{e_2} \cup \ldots \cup C_{e_{k-1}}$

Analysis

Claim: Removing C results in k components

Easy exercise

Analysis

Theorem: $w(C) \leq 2(1-1/k)$ OPT

Let A be an optimum cut and let $V_1, V_2, \ldots, V_k$ be the connected components

$A_i = \delta(V_i)$

As before assume wlog $w(A_1) \leq w(A_2) \ldots \leq w(A_k)$

and we have $w(A_1) + \ldots + w(A_k) = 2w(A) = 2\text{OPT}$

Analysis

Lemma: $\sum_{i=1}^{k-1} u(e_i) \leq \sum_{i=1}^{k-1} w(A_i)$

Assuming lemma we have

$$w(C) = u(e_1) + \ldots + u(e_{k-1})$$
$$\leq (1-1/k)(w(A_1) + \ldots + w(A_k))$$
$$\leq 2(1-1/k)\ \mathrm{OPT}$$

Proof of Lemma

We identify distinct edges $f_1, f_2, \ldots, f_{k-1}$ of T s.t $w(A_i) \geq u(f_i)$ for $1 \leq i \leq k-1$

Since algorithm picks lightest $k-1$ edges we have the lemma

Note that $f_1, \ldots, f_{k-1}$ are not necessarily related to $e_1, \ldots, e_{k-1}$

Proof of Lemma

We identify edges $f_1, f_2, \ldots, f_{k-1}$ of T s.t
$w(A_i) \geq u(f_i)$ for $1 \leq i \leq k-1$

Obtain tree T' from T as follows:

- shrink each V_i to a single vertex
- throw out parallel edges between vertices in T'

T' is connected since T is connected

Proof of Lemma

Let t_1, t_2, ..., t_k be vertices of T’ with t_i corresponding to V_i

Root T’ at t_k (recall A_k was the heaviest cut)

Orient edges in T’ towards the root

Let f_i be the unique edge directed from t_i towards the root in the orientation

Proof of Lemma

Root T' at t_k (recall A_k was the heaviest cut)

Orient edges in T' towards the root

Let f_i be the unique edge out of t_i in the orientation

Remark: f_i is an edge of T

Let $f_i = ab$ where $a \in V_i$, $b \notin V_i$

Proof of Lemma

Root T' at t_k (recall A_k was the heaviest cut)

Orient edges in T' towards the root

Let f_i be the unique edge out of t_i in the orientation

Let $f_i = ab$ where $a \in V_i$, $b \notin V_I$

From Gomory-Hu tree property $mc(ab) = u(f_i)$

Also A_i is a cut that separates a from b hence $w(A_i) \geq mc(ab) = u(f_i)$

Tight Example

Same as that for multiway cut

The Steiner k-Cut problem

Generalizes Multiway Cut and k-Cut

Given $G = (V, E)$, $w: E \rightarrow \mathcal{R}^+$

$T \subseteq V$: terminals

integer k, $k \leq |T|$

Goal: find min-wt set of edges to remove such that G is partitioned into k components each of which contains at least one terminal from T

The Steiner k-Cut problem

Given $G = (V, E)$, $w: E \rightarrow \mathcal{R}^+$

$T \subseteq V$: terminals

integer k, $k \leq |T|$

Goal: find min-wt set of edges to remove such that G is partitioned into k components each of which contains at least one terminal from T

$k = |T|$ gives multiway cut problem

$T = V$ gives k-Cut problem

Greedy Splitting/Gomory-Hu tree algs

Greedy splitting naturally defined for the problem

Difficult to analyze directly but yields 2-2/k approximation

Gomory-Hu tree based algorithm

Pick $e_1, e_2, \ldots, e_{k-1}$ from T *iteratively* such that each new edge creates a new component with a terminal. Among possible edges choose one of min weight

Theorem: 2-2/k approx for Steiner k-Cut

More general problem

Now instead of edges we allow arbitrary submodular functions

Given V and a function $f : 2^V \rightarrow \mathcal{R}^+$

$T \subseteq V$

integer $k \leq |T|$

Goal: partition V into $V_1, V_2, \ldots, V_k$ such that $V_i \cap T \neq \emptyset$ for $1 \leq i \leq k$ so as to minimize $\sum_i f(V_i)$

More general problem

Given V and a function $f : 2^V \rightarrow \mathcal{R}^+$

$T \subseteq V$

integer $k \leq |T|$

Goal: partition V into $V_1, V_2, \ldots, V_k$ such that $V_i \cap T \neq \emptyset$ for $1 \leq i \leq k$ so as to minimize $\sum_i f(V_i)$

Theorem: If f is *submodular* and *symmetric* Greedy splitting yields a $2-2/k$ approximation

(also for some other cases of f)

More general problem

Theorem: If f is *submodular* and *symmetric*
Greedy splitting yields a 2-2/k approximation

(also for some other cases of f)

If G is a graph or a hypergraph then the cut function $f(V_i) = w(\delta(V_i))$ is a symmetric submodular function

Hence Steiner k-cut is a special case

Why does it work?

Direct proof using submodularity [Zhao-Nagamochi-Ibaraki'05]

Another proof:

If f is submodular and symmetric then it also admits a Gomory-Hu tree!

In other words, Gomory-Hu tree exists because cut-functions in graphs are submodular and symmetric

Greedy Splitting for Steiner k-Cut

We adapt proof of [Zhao-Nagamochi-Ibaraki'05] for submodular splitting problems and simplify it for the Steiner k-Cut problem

Some notation
Greedy generations vertex partitions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{k-1}$ starting with $\mathcal{P}_0 = \{V\}$

$\mathcal{P}_i$ is a refinement of $\mathcal{P}_{i-1}$ with one of the components of $\mathcal{P}_{i-1}$ split into two components

Greedy Splitting for Steiner k-Cut

Some notation
Greedy generations vertex partitions $\mathcal{P}_1$, $\mathcal{P}_2$, ..., $\mathcal{P}_{k-1}$ starting with $\mathcal{P}_0 = \{V\}$

$\mathcal{P}_i$ is a refinement of $\mathcal{P}_{i-1}$ with one of the components of $\mathcal{P}_{i-1}$ split into two components

We are interested only in partitions that are *valid*: that is each component contains a terminal

Analysis of Greedy Splitting

Let $w(\mathcal{P}_i)$ the cost of edges cut in the partition $\mathcal{P}_i$

We prove the following by induction on i

Lemma:
For *any* valid partition $\mathcal{P} = \{V_1, V_2, \ldots, V_i\}$
$w(\mathcal{P}_{i-1}) \leq w(\delta(V_1)) + w(\delta(V_2)) + \ldots + w(\delta(V_{i-1}))$

(note that rhs doesn't depend on V_i)

Analysis of Greedy Splitting

For *any* valid partition $\mathcal{P} = \{V_1, V_2, \ldots, V_i\}$

$w(\mathcal{P}_{i-1}) \leq w(\delta(V_1)) + w(\delta(V_2)) + \ldots + w(\delta(V_{i-1}))$

Suppose above is true: consider an optimum solution for problem $\{A_1, A_2, \ldots, A_k\}$ where the ordering is chosen s.t

$w(\delta(A_k)) \geq w(\delta(A_i))$ for $1 \leq i \leq k-1$

Analysis of Greedy Splitting

Suppose above is true: consider an optimum solution for problem $\{A_1, A_2, \ldots, A_k\}$ where the ordering is chosen s.t

$w(\delta(A_k)) \geq w(\delta(A_i))$ for $1 \leq i \leq k\text{-}1$

From the lemma with $i = k\text{-}1$,
$w(\mathcal{P}_{k-1}) \leq (1\text{-}1/k)\,(w(\delta(A_1)) + \ldots + w(\delta(A_k)))$
$\leq 2(1\text{-}1/k)\ \mathrm{OPT}$

Proof of the lemma

Base case $i = 1$ is easy to check

Assume hypothesis holds for $i-1$

Let $\mathcal{P} = (V_1, V_2, \ldots, V_i)$ be an arbitrary *valid* partition into i components

Consider $\mathcal{P}_{i-2}$:

There must be some component $W \in \mathcal{P}_{i-2}$ and two indices $h < l$ with $W \cap V_h$ *and* $W \cap V_l$ both containing a terminal (Why?)

Proof of the lemma

Therefore splitting W into $W \cap V_h$ and $W - V_h$

would be considered in the Greedy Splitting algorithm when refining $\mathcal{P}_{i-2}$ into $\mathcal{P}_{i-1}$

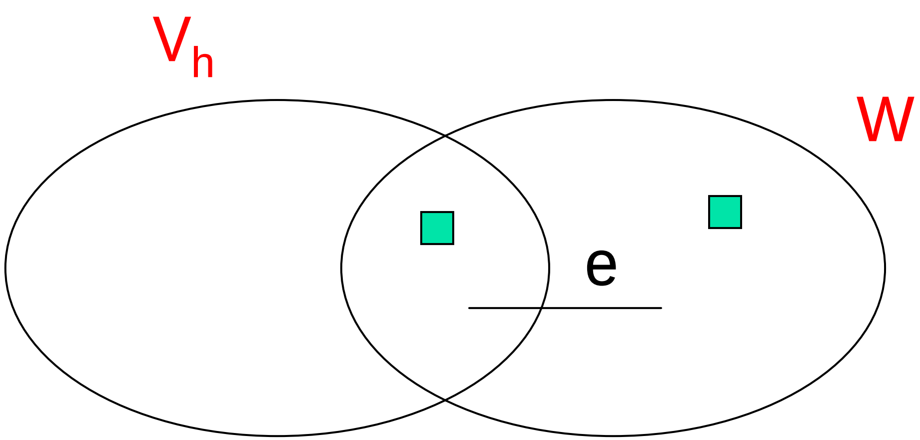
Since Greedy chose a cheapest split

$w(\mathcal{P}_{i-1}) - w(\mathcal{P}_{i-2}) \leq$ increase in cost if Greedy chose to split W as above

We claim that splitting W as above increases cost by at most $w(\delta(V_h))$

Proof of the lemma

We claim that splitting W as above increases cost by at most $w(\delta(V_h))$



Any new edge e induced by the split of W is from $(V_h \cap W)$ to $(W - V_h)$ so must be in $\delta(V_h)$

Proof of lemma

Therefore $w(\mathcal{P}_{i-1}) - w(\mathcal{P}_{i-2}) \leq w(\delta(V_h))$

Apply induction hypothesis on $i-1$ to the partition

$\mathcal{P}' = \{V_1,\ldots,V_{h-1}, V_{h+1},\ldots,V_{i-1}, V_h \cup V_i\}$

(we removed V_h and merged it with V_i, use the fact that $h < l \leq i$)

Proof of lemma

Therefore $w(\mathcal{P}_{i-1}) - w(\mathcal{P}_{i-2}) \leq w(\delta(V_h))$

Apply induction hypothesis on i-1 to the partition
$\mathcal{P}' = \{V_1,\ldots,V_{h-1}, V_{h+1},\ldots,V_{i-1}, V_h \cup V_i\}$

$w(\mathcal{P}_{i-2}) \leq (\sum_{j=1}^{i-1} w(\delta(V_j))) - w(\delta(V_h))$ (since V_h is no longer there in $\mathcal{P}'$)

Combining above two inequalities proves the hypothesis for i

Exercises/Reading

Read about Gomory-Hu tree construction in Vazirani's book, Chapter 4 exercise

Prove that the Greedy Split algorithm is optimal when G is a tree

Try to extend the analysis of the Gomory-Hu tree based algorithm for k-Cut to the algorithm for the Steiner k-Cut problem