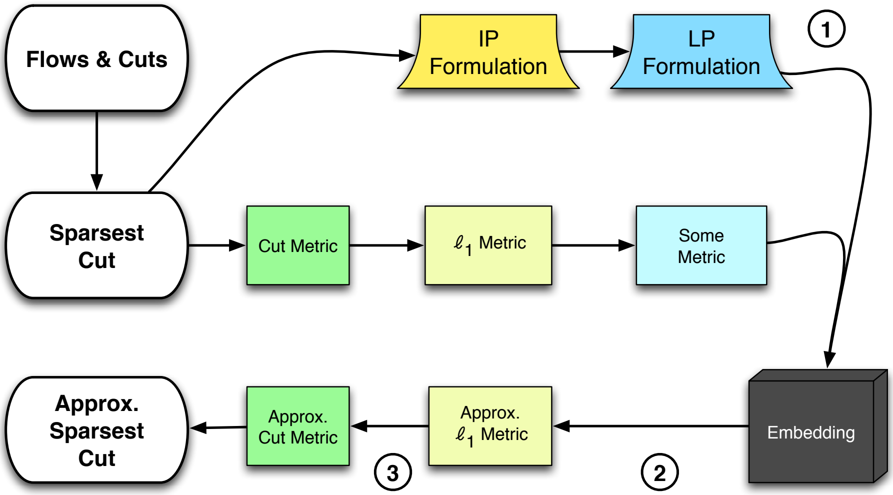
Sparsest Cut

Group 5: Daryl, Etkin, Rajendra, Aarthi and Supartha

National University of Singapore

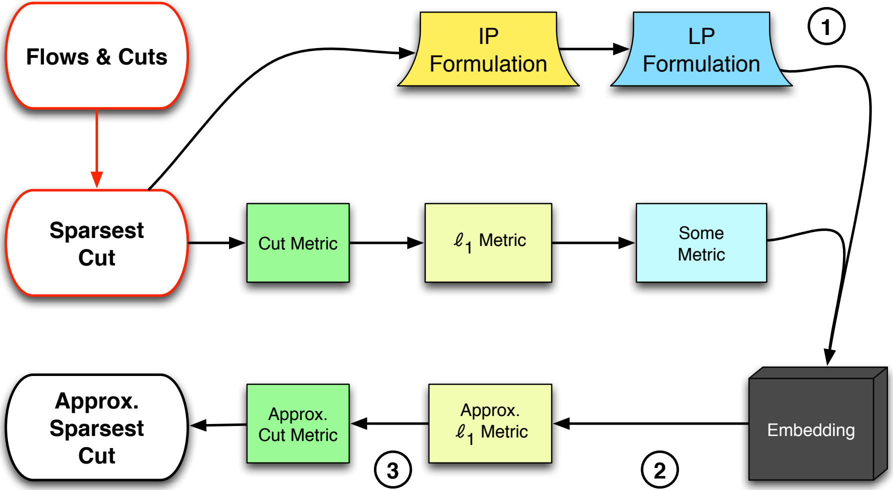
April 12, 2013

Roadmap



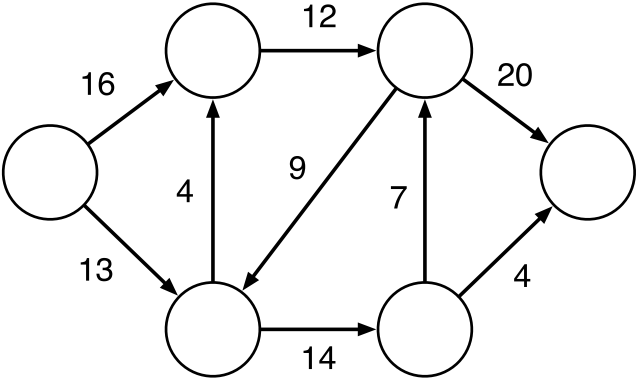
Flows & Cuts
IP Formulation
LP Formulation
1
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Embedding
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Embedding
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2

Flow Networks

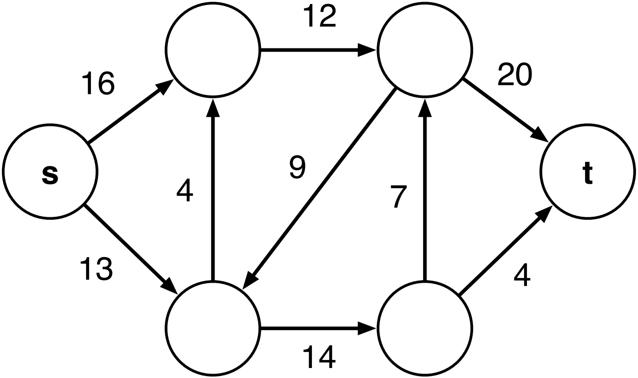


Directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$.

Nodes may be **sources** or **sinks** of a flow.

Maximum s-t Flow Problem

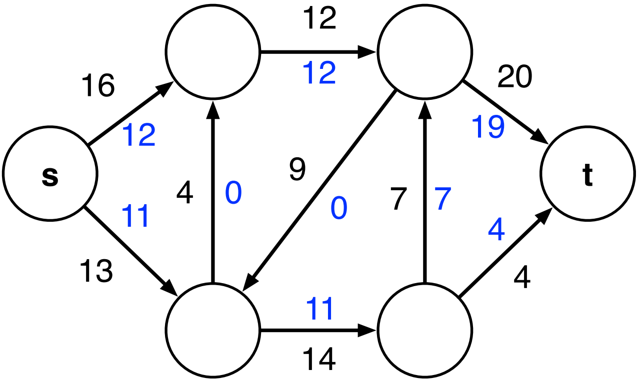
A **flow** is an assignment to each edge $f : E \to \mathbb{R}^+$ subject to capacity constraints and flow conservation.



Given source s and sink t, find the maximum possible flow.

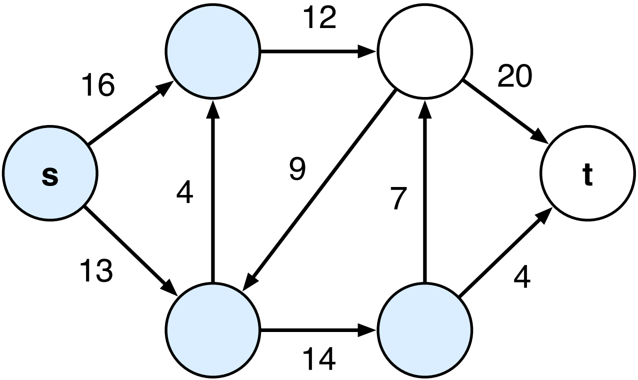
Maximum s-t Flow Problem

A **flow** is an assignment to each edge $f : E \to \mathbb{R}^+$ subject to capacity constraints and flow conservation.



Given source s and sink t, find the maximum possible flow.

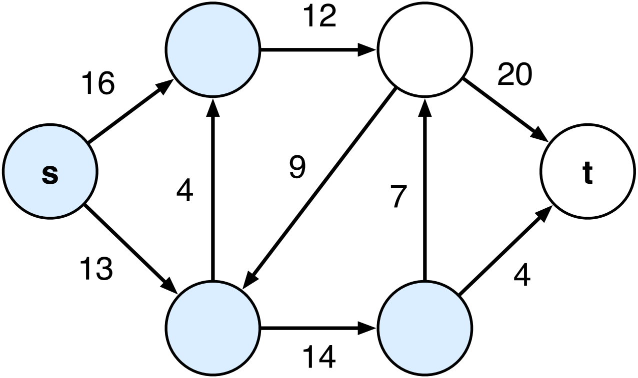
Cuts of Flow Networks



A **cut** S is a partitioning of vertices V into S and $T = V - S$ such that $s \in S$ and $t \in T$.

Minimum s-t Cut Problem

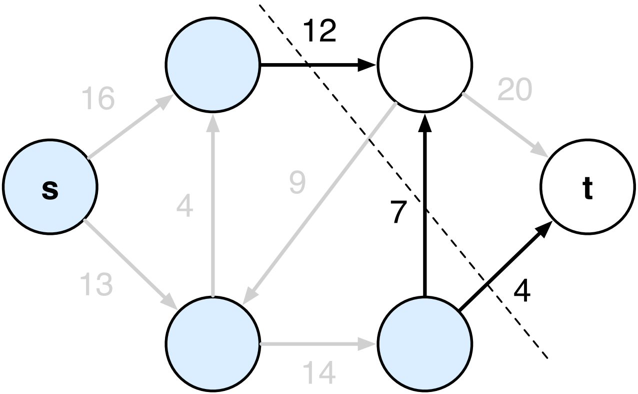
The **capacity** of a cut S is the sum of edge capacities from S to T.



Find a cut with minimum capacity.

Minimum s-t Cut Problem

The **capacity** of a cut S is the sum of edge capacities from S to T.



Find a cut with minimum capacity.

Ford-Fulkerson Algorithm (1954)

- Maximum flow problems are related to minimum cut problems
- Ford-Fulkerson algorithm solves *both* maximum flow and minimum cut simultaneously in polynomial time

Theorem (Max-Flow Min-Cut)

The maximum flow between a source s and sink t
is equal to
the minimum capacity over all cuts separating s and t.

Ford-Fulkerson Algorithm (1954)

- Maximum flow problems are related to minimum cut problems
- Ford-Fulkerson algorithm solves *both* maximum flow and minimum cut simultaneously in polynomial time

Theorem (Max-Flow Min-Cut)

The maximum flow between a source s and sink t
is equal to
the minimum capacity over all cuts separating s and t.

Does this result apply to multiple-commodities?

Max-Flow Min-Cut for Multiple Commodities

- **Commodities:** independent flows that cannot mix but must still share network capacity, e.g.
    - water and oil through pipes
    - individual TCP connections over the Internet
    - various shipments through a road network
- **NOT** as simple as multiple sources and multiple sinks
    - Example: a flow of water from some source should not end up at a sink for oil
- The strong Max-Flow Min-Cut Theorem only applies to *single-commodity* flow/cut problems

Two Multi-Commodity Generalizations

1. Maximum Multi-Commodity Flow
   - Minimum Multi-Cut
2. Maximum Concurrent Multi-Commodity Flow
   - Sparsest Cut

Two Multi-Commodity Generalizations

1. Maximum Multi-Commodity Flow (LP)
    - Minimum Multi-Cut (NP-Hard)
2. Maximum Concurrent Multi-Commodity Flow (LP)
    - Sparsest Cut (NP-Hard)

Two Multi-Commodity Generalizations

1. Maximum Multi-Commodity Flow (LP)
   - Minimum Multi-Cut (NP-Hard)
2. Maximum Concurrent Multi-Commodity Flow (LP)
   - Sparsest Cut (NP-Hard)

* Typically defined with respect to *undirected* graphs; the problems are much harder in directed graphs.

Maximum Concurrent Multi-Commodity Flow

a.k.a. Demands Multi-Commodity Flow

Undirected Graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$.

For k commodities, let $\left\{(s_1, t_1), ..., (s_k, t_k)\right\}$ be the set of source-sink pairs for each commodity.

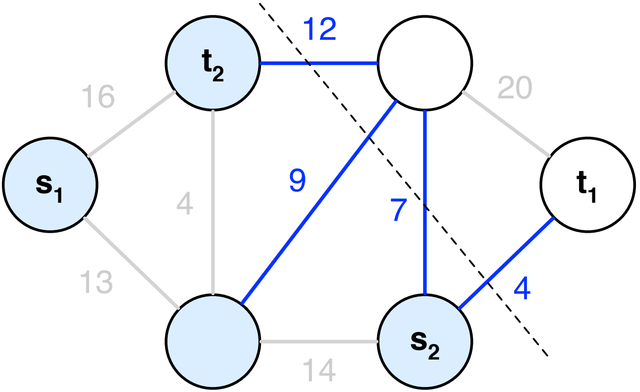
The **demand** for the commodities is a function $d : \{1...k\} \to \mathbb{R}^+$.

The Flow Problem

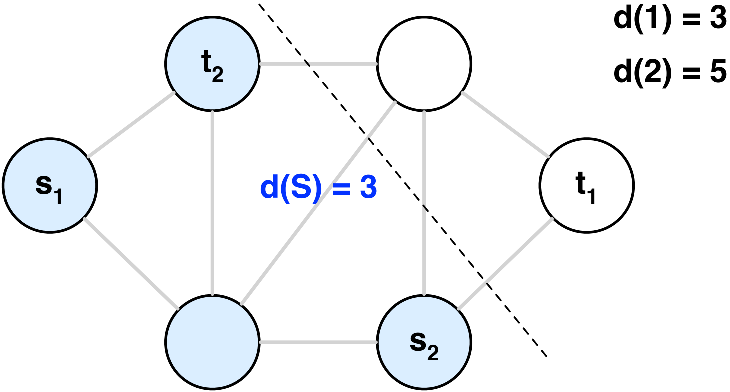
Find a flow that maximizes **throughput** α, where $\alpha.d(i)$ units of each commodity i are flowing *simultaneously*.

Demand constraint ensures "fairness".
Commodities must flow in the specified demand ratio.

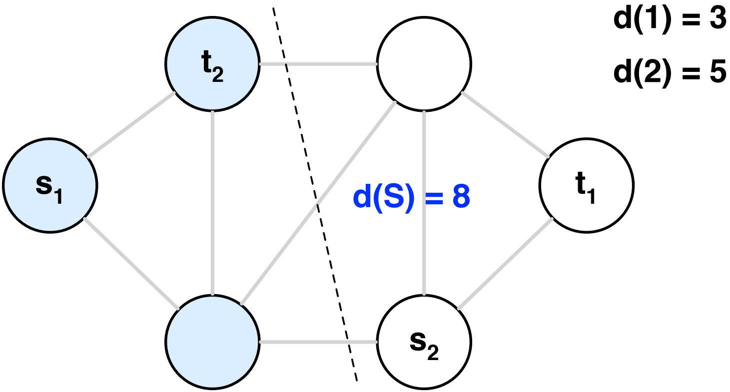
Define **capacity of a cut** $c(S)$ to be the sum of edge capacities between partitions S and $S - V$.



Define **demand of a cut** $d(S)$ to be the sum of the demands of commodity pairs separated by the cut.



Define **demand of a cut** $d(S)$ to be the sum of the demands of commodity pairs separated by the cut.



Given *any* valid flow with throughput α and *any* cut S,

$$\alpha.d(S) \leq c(S)$$

Consequently, for optimal throughput $\alpha*$,

$$\alpha* \leq \min_{S \subset V} \frac{c(S)}{d(S)}$$

Same undirected graph with edge capacities,
k source-sink commodity pairs and demands.

Define the **sparsity of a cut** to be $\Phi(S) = \frac{c(S)}{d(S)}$.

The Cut Problem

Find a cut S of minimum sparsity $\Phi(S)$.

This problem is **NP-Hard**. *[Matula & Shahrokhi '90]*

Recall that minimum sparsity puts the most stringent upper-bound on maximum throughput...

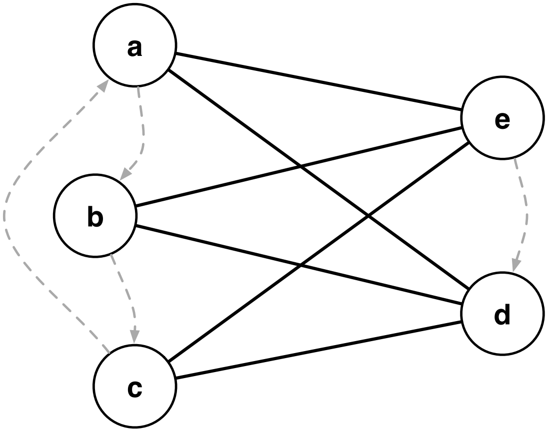
$$\alpha* \leq \min_{S \subset V} \Phi(S)$$

Are min sparsity and max throughput equivalent?

- $k = 1$ (Single-Commodity): **Yes!** *[Ford-Fulkerson '54]*
- $k > 1$ (Arbitrary): **Not necessarily.**

Counter-Example

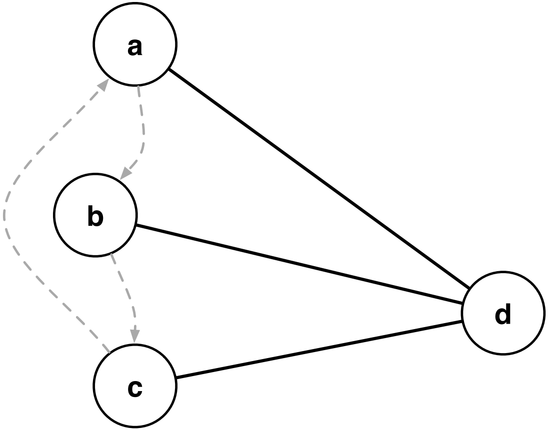
Network with 4 commodity pairs (dotted lines) and 6 edges (solid lines) with unit capacities and unit demands.



Minimum sparsity is 1. (You can verify this.)

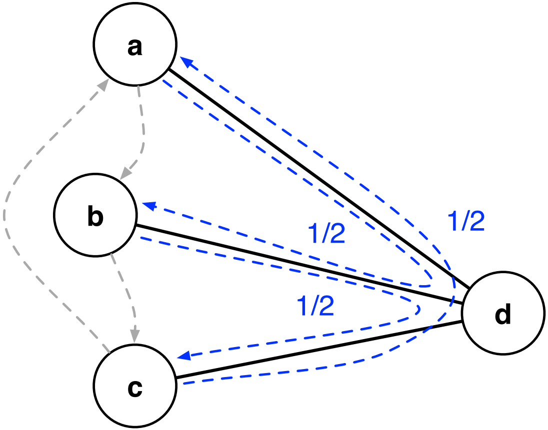
Counter-Example

Ignore node e and its associated edges and commodities.



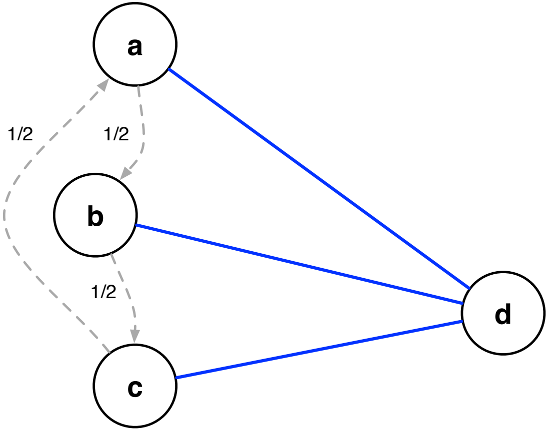
Counter-Example

Start fulfilling demands of first 3 commodities with 3 edges.



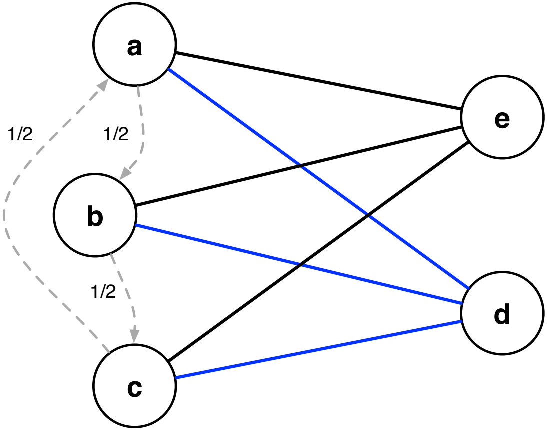
Counter-Example

Edges fully saturated, and half of each commodity's demands are met.



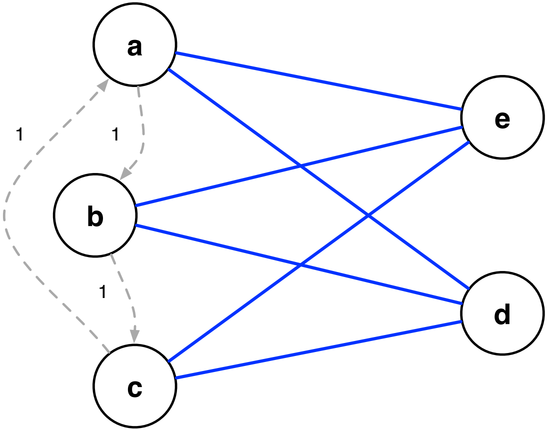
Counter-Example

Re-introduce node *e* and the remaining edges.



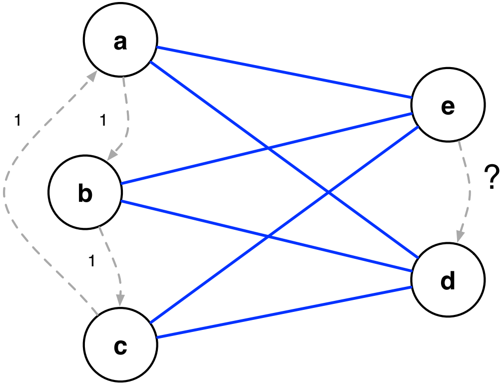
Counter-Example

Repeat assignment. Demands of first 3 commodities fully met.



Counter-Example

No more capacity to route last commodity!



Minimum sparsity 1, but maximum throughput strictly less than 1.

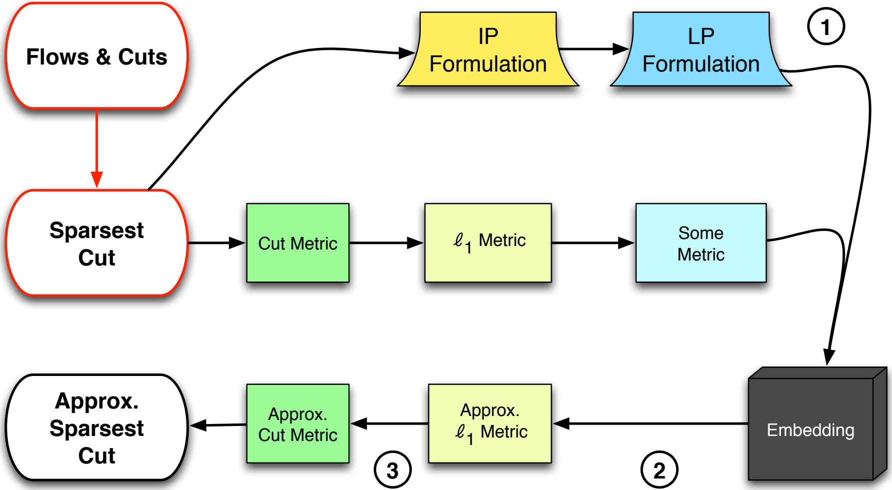
Without strong duality, cannot simply use max-flow solution to solve Sparsest Cut in polynomial time.

BUT minimum sparsity cannot be arbitrarily larger than maximum throughput.

Approximation algorithms put a bound on their ratio.

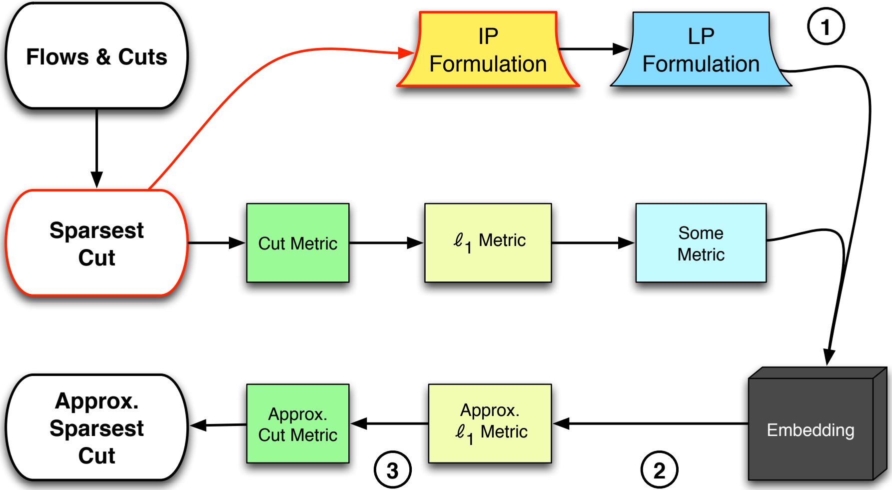
We can still derive *approximate* max-flow min-cut theorems for multi-commodity flows.

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2
Embedding

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2
Embedding

Applications

- Building Robust Computer Networks
- Balanced Cut
- Edge Expansion
- Conductance
- Minimum Cut Linear Arrangement

What is an Approximation Algorithm?

- Used to find approximate solution to optimization problems
- Especially for NP-hard problems (no polynomial time solution)
- Better option than heuristics
  - Provable solution quality and run-time bound
- Also being used for problems with large input size, although has a known polynomial-time algorithms

Related Work

Name	Graphs	Approximation	Based on
Leighton et al. 1988	restricted	$O(\log n)$	LP
Klein et al. 1995	general	$O(\log C \log D)$	LP
Linial et al. 1995	general	$O(\log k)$	LP
Arora et al. 2008	general	$O(\sqrt{\log k} \log \log k)$	SDP
Chakrabarti et al. 2008	restricted	$O(1)$	LP
Chekuri et al. 2010	restricted	$O(1)$	LP

We will be covering the $O(\log k)$ algorithm by Linial et al.

Sparsity Ratio

- Graph: $G = (V, E)$
- Cost: $c(e)$, edge: $e \in E$
- Vertex pair: (s_i, t_i), $i = 1, ..., k$

- $I(S) = \{i : |S \cap \{s_i, t_i\}| = 1\}$: terminal pairs that are disconnected by S, where S is a subset of vertices ($S \subseteq V$)
- Removed edges by S: $\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$

- Sparsity ratio of S:
  $\Phi(S) = \frac{\sum_{e \in \delta(s)} c(e)}{\sum_{i \in I(S)} d(i)} = \frac{total\ \ capacity\ \ of\ \ edges\ \ removed\ \ by\ \ S}{total\ \ demand\ \ of\ \ commodities\ \ disconnected\ \ by\ \ S}$

Sparsity Ratio

- We extend the sparsity ratio to apply all removed edges F
- We denote the new graph as $\bar{G} = (V, E - F)$
- Set of connected components:
  $\mathcal{S} = \{S_1, S_2, ..., S_c\} = \{i : s_i \in S_j, t_i \in S_k, j \neq k\}$
- Sparsity ratio of S:
  $\Phi(\mathcal{S}) = \frac{\sum_{e \in F} c(e)}{\sum_{i \in I(S)} d(i)} = \frac{total\ \ capacity\ \ of\ \ all\ \ removed\ \ edges}{total\ \ demand\ \ of\ \ all\ \ disconnected\ \ commodities}$
- $min_{S_i \in \mathcal{S}} \Phi(S_i) \leq \Phi(\mathcal{S})$

IP Formulation

- $\mathcal{P}_i$ denotes the set of paths for commodity i

Minimize

$$\frac{\sum_{e \in E} c(e)x(e)}{\sum_{i=1}^{k} d(i)y(i)}$$

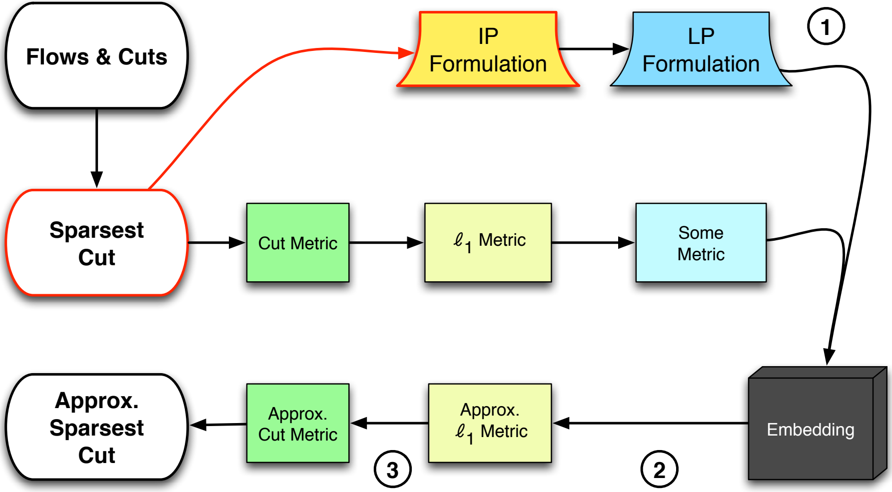
subject to

$$\sum_{e \in P} x(e) \geq y(i). \text{ for each } P \in \mathcal{P}_i, i = 1, ..., k,$$

$$y(i) \in \{0, 1\}, \text{ for each } i = 1, ..., k,$$

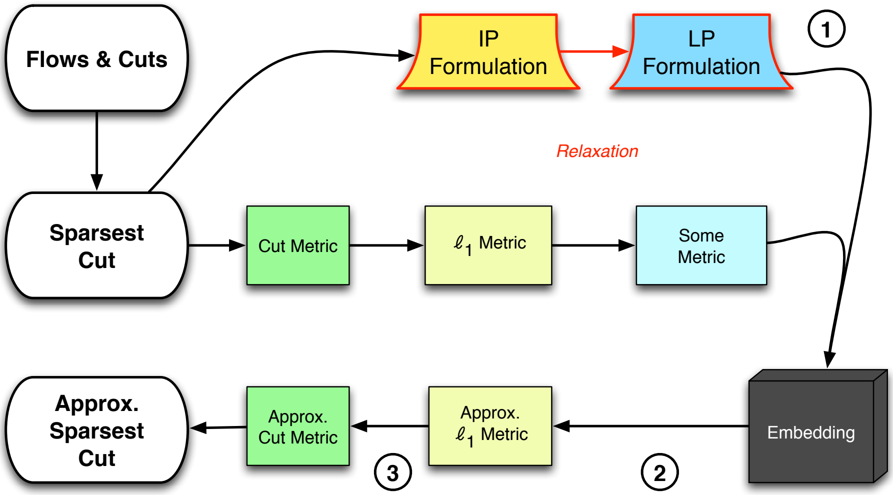
$$x(e) \in \{0, 1\}, \text{ for each } e \in E.$$

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2
Embedding

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
Embedding
3
2

LP Relaxation

- We replace
  $y(i) \in \{0, 1\}$, for each $i = 1, ..., k$,
  $x(e) \in \{0, 1\}$, for each $e \in E$.
- with
  $y(i) \geq 0$, for each $i = 1, ..., k$,
  $x(e) \geq 0$, for each $e \in E$.
- We add

  $\sum_{i=1}^{k} d(i)y(i) = 1$

LP Formulation

Minimize

$$\sum_{e \in E} c(e)x(e)$$

subject to

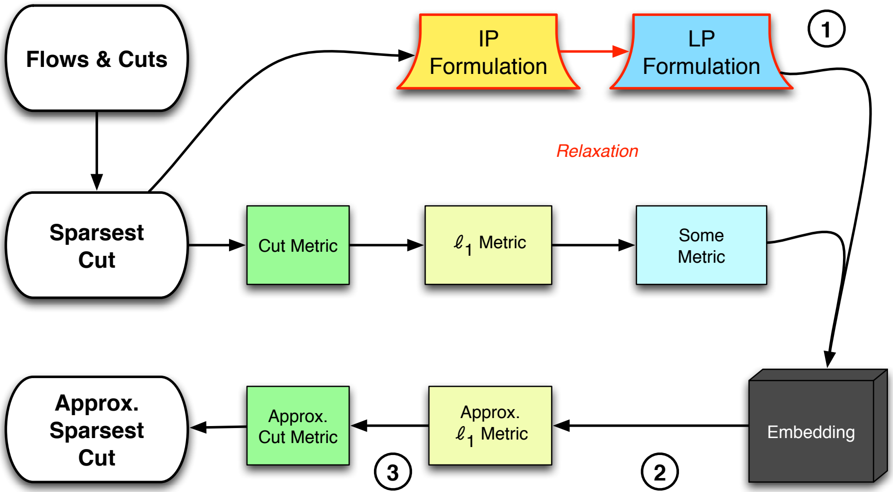
$$\sum_{i=1}^{k} d(i)y(i) = 1$$

$$\sum_{e \in P} x(e) \geq y(i). \text{ for each } P \in \mathcal{P}_i, i = 1, ..., k,$$

$$y(i) \geq 0, \text{ for each } i = 1, ..., k,$$

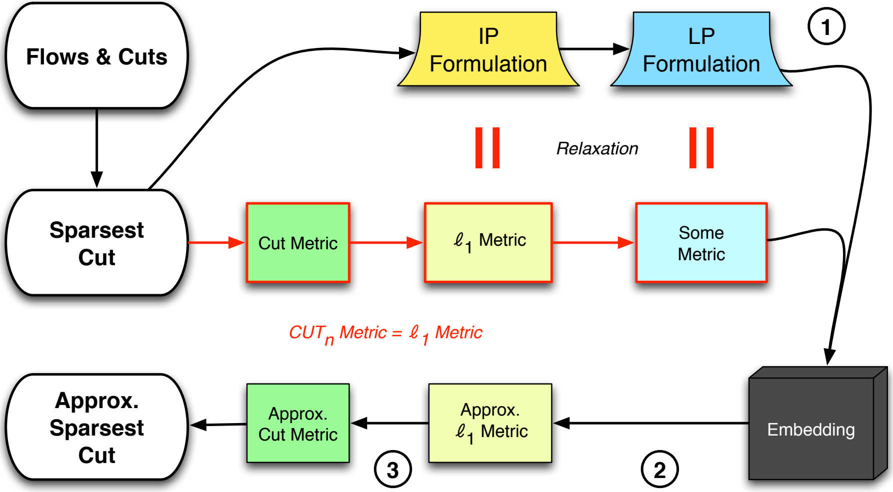
$$x(e) \geq 0, \text{ for each } e \in E.$$

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Embedding
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
3
2

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
3
Approx. ℓ_1 Metric
2
Embedding

What is a Metric

A metric on a set V is defined as a function $d : V \times V \to \mathbb{R}$

- $d(x, y) = 0$ iff $x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) + d(y, z) \geq d(x, z)$

Example: $d(x, y) = \sum_{i=1}^{m} |x_i - y_i|$ is a metric on $\mathbb{R}^m$ (called ℓ_1 metric)

Example: $d(x, y) = (\sum_{i=1}^{m} |x_i - y_i|^p)^{1/p}$ is a metric on $\mathbb{R}^m$ (called ℓ_p metric)

What is a Cut Metric

A length function induced by a subset (cut) S of V is defined as a function $\delta_S : V \times V \to \mathbb{R}$

- $\delta_S(x, y) = 0$ if $x, y \in S$ or $x, y \in \overline{S}$
- $\delta_S(x, y) = 1$ otherwise

Easy to check that cut metric is in fact a (semi) metric

Metrics as Vectors in $\mathbb{R}^{n(n-1)/2}$

- Any n-point metric can be associated with a vector in $\mathbb{R}^{n(n-1)/2}$ with each coordinate corresponding to a pair of vertices from the metric
- Set of all metrics on V forms a convex cone in $\mathbb{R}^{n(n-1)/2}$.

Definition of Cone: If d_1, d_2 are in $\mathbb{R}^{n(n-1)/2}$ then $\alpha d_1 + \beta d_2 \in \mathbb{R}^{n(n-1)/2}$ for non-negative reals α, β.

Sparsest Cut in $\mathbb{R}^{n(n-1)/2}$

In this setting the sparsest cut problem can be restated as

$$\min_{\text{all cut metrics } S} \frac{\overline{c} \cdot \overline{\delta_S}}{\overline{D} \cdot \overline{\delta_S}}$$

- $\overline{c}$: vector in $\mathbb{R}^{n(n-1)/2}$ with $\overline{c_{ij}}$ being the capacity of the edge between vertex i and j
- $\overline{D_{ij}}$: demand between vertex i and j
- $\overline{D} \cdot \overline{\delta_S}$ is the dot product of two corresponding vectors.

ℓ_1 and Cut Metric

Let us denote the positive cone generated by all cut metrics by CUT_n.

$$CUT_n = \{\overline{d} | d = \textstyle\sum_{S \subset V} \alpha_S \delta_S,\ \alpha_S > 0\ \forall\ S\}$$

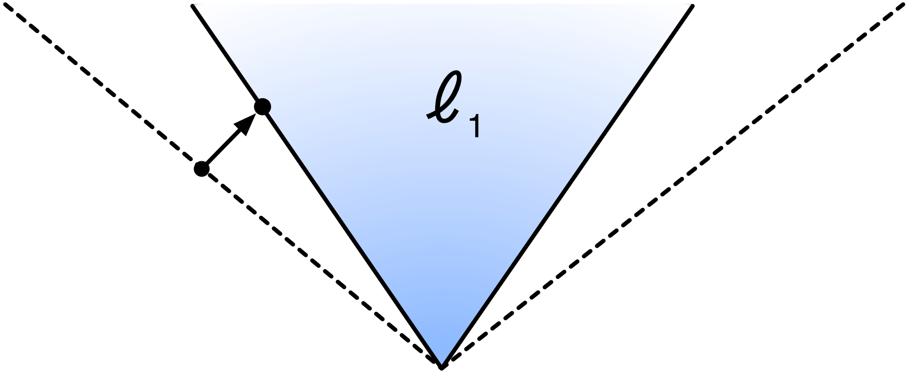
So the optimum to the above formulation will be achieved at some extreme point on the cone.

$$\Phi^* = \min_{d \in CUT_n} \frac{\overline{c} \cdot \overline{\delta_S}}{\overline{D} \cdot \overline{\delta_S}}$$

Now we can claim:

- Cut metrics are the extreme rays of the cone of ℓ_1 metrics
- CUT_n = set of all ℓ_1 metrics

Metric Cones



Consider any metric in CUT_n.

For every S with $\alpha_S > 0$, we have a dimension and in that dimension we put value 0 for $x \in S$ and value $\alpha_S$for $x \in \overline{S}$.

Hence $CUT_n \subset \ell_1$ metrics.

Consider a set of n points from $\mathbb{R}^n$.

Take one dimension d and sort the points in increasing value along that dimension. Say we get $v_1, v_2, ..v_k$ as as the set of distinct values.

Define $k-1$ cut metrics $S_i = \{x | x_d \leq v_{i+1}\}$ and let $\alpha_i = v_{i+1} - v_i$. Now along this dimension, $|x_d - y_d| = \sum_{i=1}^{k} \alpha_i \delta_{S_i}$.

We can construct cut metrics for every dimension. Hence we have a metric in CUT_n for every n-point metric in ℓ_1.

Given a ℓ_1 metric μ in $\mathbb{R}^{\mathcal{D}}$
we can decompose $\mu = \sum_{S \subset V} \alpha_S \delta_S$
to at most $n\mathcal{D}$ cut metrics where $\alpha_S \geq 0$.

Sparsest Cut over ℓ_1

Now the formulation for sparsest cut can be written as

$$\Phi^* = \min_{d \in \ell_1 metrics} \frac{\overline{c} \cdot \overline{\delta_S}}{\overline{D} \cdot \overline{\delta_S}}$$

But as sparsest cut is NP-hard, we cannot hope to solve over ℓ_1 metrics. Hence we consider a relaxation of this problem to the domain of set of all metrics

$$\Lambda^* = \min_{d \in all\ metrics} \frac{\overline{c} \cdot \overline{d}}{\overline{D} \cdot \overline{d}}$$

Clearly,

$$\Lambda^* \leq \Phi^*.$$

We can solve for Λ^* using a linear program

- min $\sum c_{ij} d_{ij}$
- subject to:
  - $d_{ij} \leq d_{ik} + d_{kj}$
  - $\sum D_{ij} d_{ij} = 1$
  - $d_{ij} \geq 0$

Bound on the Integrality Gap

Solve the LP to find d that achieves Λ^*.

Embed d in to ℓ_1 metrics with low distortion.

Get a cut metric from the ℓ_1 metric.

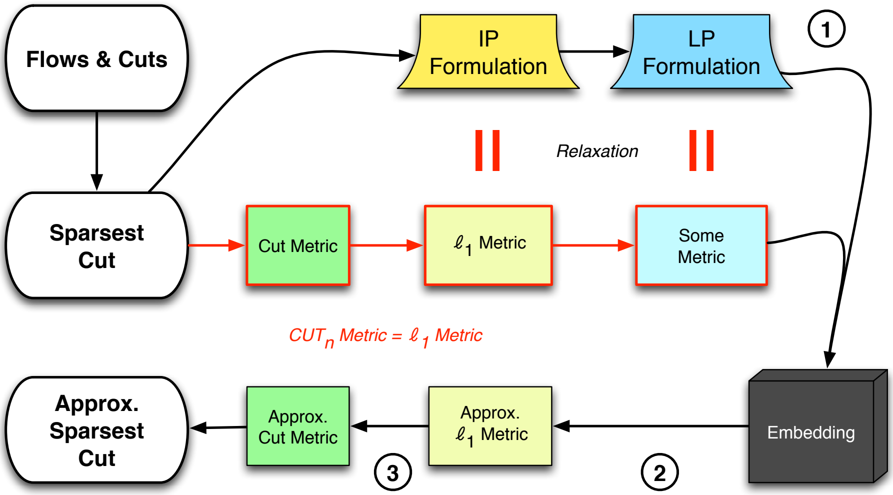
Depending on the distortion the we can bound the integrality gap.

Result: Suppose for each metric d there exist a ℓ_1 metric μ such that

$$d(x,y) \leq \mu(x,y) \leq \alpha d(x,y), \text{ for all } x,y \in V$$

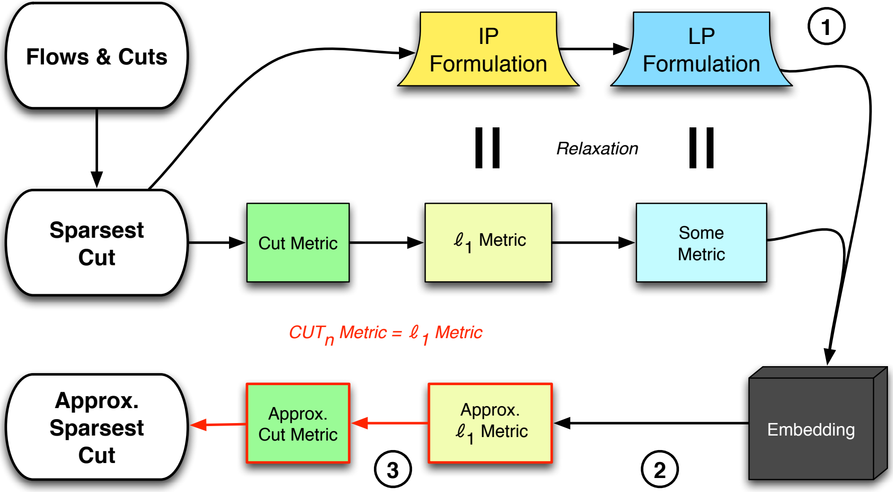
Then sparsest cut LP has integrality gap at most α.

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
Embedding
3
2

Roadmap


Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
Embedding
3
2

Obtaining a Cut from an ℓ_1 metric

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$
- Repeat for all dimensions $i = 1 \ldots \mathcal{D}$

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$
- Repeat for all dimensions $i = 1 \dots \mathcal{D}$
  - Sort vertices $v \in V$ in increasing order of coordinates in dimension i

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$
- Repeat for all dimensions $i = 1 \ldots \mathcal{D}$
  - Sort vertices $v \in V$ in increasing order of coordinates in dimension i
  - Number them accordingly so that we have $f_i(v_1) \leq f_i(v_2) \leq \ldots f_i(v_n)$

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$
- Repeat for all dimensions $i = 1 \ldots \mathcal{D}$
  - Sort vertices $v \in V$ in increasing order of coordinates in dimension i
  - Number them accordingly so that we have $f_i(v_1) \leq f_i(v_2) \leq \ldots f_i(v_n)$
  - Repeat for all $j = 1 \ldots n$
    - Create Set $S_{ij} = \{v_k | v's \text{ are sorted according to dim } i; 1 \leq k \leq j\}$
    - Calculate the sparsity ratio for $S_{ij} = \Phi(S_{ij})$

Obtaining a Cut from an ℓ_1 metric

- Given a metric $d \in \ell_1$ in space $\mathbb{R}^{\mathcal{D}}$
- Repeat for all dimensions $i = 1 \ldots \mathcal{D}$
  - Sort vertices $v \in V$ in increasing order of coordinates in dimension i
  - Number them accordingly so that we have $f_i(v_1) \leq f_i(v_2) \leq \ldots f_i(v_n)$
  - Repeat for all $j = 1 \ldots n$
    - Create Set $S_{ij} = \{v_k | v's \text{ are sorted according to dim } i; 1 \leq k \leq j\}$
    - Calculate the sparsity ratio for $S_{ij} = \Phi(S_{ij})$
- Take the cut S_{ij} which has the minimum sparsity ratio, giving the required approximate cut.

Obtaining a Cut from an ℓ_1 metric

Continued...

For $d \in \ell_1$, from the previous algorithm we have the representation:
$d = \sum_{S \in \mathcal{S}} \alpha_S \delta_S$.
Consider ϕ to be the sparsity ratio obtained from the values of d. So,

Obtaining a Cut from an ℓ_1 metric

Continued...

For $d \in \ell_1$, from the previous algorithm we have the representation:
$d = \sum_{S \in \mathcal{S}} \alpha_S \delta_S$.
Consider ϕ to be the sparsity ratio obtained from the values of d. So,

$$\phi = \frac{\sum_{e \in E} c(e) x(e)}{\sum_{i=1}^{k} d(i) y(i)}$$

Obtaining a Cut from an ℓ_1 metric

Continued...

For $d \in \ell_1$, from the previous algorithm we have the representation: $d = \sum_{S \in \mathcal{S}} \alpha_S \delta_S$.
Consider ϕ to be the sparsity ratio obtained from the values of d. So,

$$\begin{aligned}\phi &= \frac{\sum_{e \in E} c(e) x(e)}{\sum_{i=1}^{k} d(i) y(i)} \\ &= \frac{\sum_{e \in E} c(e) \sum_{S \in \mathcal{S}} \alpha_S \delta_S(e)}{\sum_i d(i) \sum_{S \in \mathcal{S}} \alpha_S \delta_S(s_i, t_i)}\end{aligned}$$

Obtaining a Cut from an ℓ_1 metric

Continued...

For $d \in \ell_1$, from the previous algorithm we have the representation:
$d = \sum_{S \in \mathcal{S}} \alpha_S \delta_S$.
Consider ϕ to be the sparsity ratio obtained from the values of d. So,

$$
\begin{aligned}
\phi &= \frac{\sum_{e \in E} c(e) x(e)}{\sum_{i=1}^{k} d(i) y(i)} \\
&= \frac{\sum_{e \in E} c(e) \sum_{S \in \mathcal{S}} \alpha_S \delta_S(e)}{\sum_i d(i) \sum_{S \in \mathcal{S}} \alpha_S \delta_S(s_i, t_i)} \\
&= \frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in E} c(e) \delta_S(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_i d(i) \delta_S(s_i, t_i)}
\end{aligned}
$$

Contd...

Obtaining a Cut from an ℓ_1 metric

Continued...

By definition, $\delta_S(u, v) = 1$ only when $e = (u, v)$ crosses the cut defined by S i.e. $e \in \delta(S)$. This means,

$$\frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in E} c(e) \delta_S(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_i d(i) \delta_S(s_i, t_i)}$$

Obtaining a Cut from an ℓ_1 metric

Continued...

By definition, $\delta_S(u, v) = 1$ only when $e = (u, v)$ crosses the cut defined by S i.e. $e \in \delta(S)$. This means,

$$\frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in E} c(e) \delta_S(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_i d(i) \delta_S(s_i, t_i)} = \frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in \delta(S)} c(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_{i \in \mathcal{I}(S)} d(i)}$$

Obtaining a Cut from an ℓ_1 metric

Continued...

By definition, $\delta_S(u,v) = 1$ only when $e = (u,v)$ crosses the cut defined by S i.e. $e \in \delta(S)$. This means,

$$
\begin{aligned}
\frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in E} c(e)\delta_S(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_i d(i)\delta_S(s_i, t_i)} &= \frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in \delta(S)} c(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_{i \in \mathcal{I}(S)} d(i)} \\
&\geq \min_{S \in \mathcal{S}} \frac{\alpha_S \sum_{e \in \delta S} c(e)}{\alpha_S \sum_{i \in \mathcal{I}(S)} d(i)}
\end{aligned}
$$

Obtaining a Cut from an ℓ_1 metric

Continued...

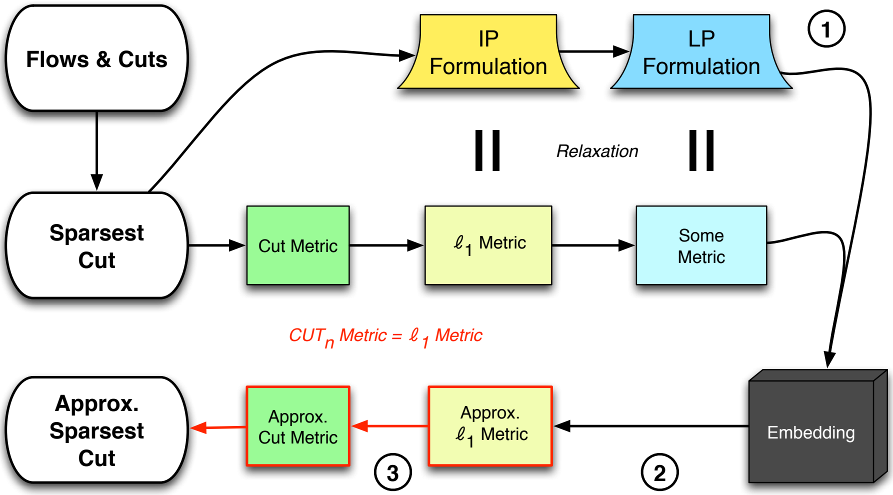
By definition, $\delta_S(u,v) = 1$ only when $e = (u,v)$ crosses the cut defined by S i.e. $e \in \delta(S)$. This means,

$$
\begin{aligned}
\frac{\sum_{S\in\mathcal{S}} \alpha_S \sum_{e\in E} c(e)\delta_S(e)}{\sum_{S\in\mathcal{S}} \alpha_S \sum_i d(i)\delta_S(s_i,t_i)} &= \frac{\sum_{S\in\mathcal{S}} \alpha_S \sum_{e\in\delta(S)} c(e)}{\sum_{S\in\mathcal{S}} \alpha_S \sum_{i\in\mathcal{I}(S)} d(i)} \\
&\geq \min_{S\in\mathcal{S}} \frac{\alpha_S \sum_{e\in\delta S} c(e)}{\alpha_S \sum_{i\in\mathcal{I}(S)} d(i)} \\
&= \min_{S\in\mathcal{S}} \frac{\sum_{e\in\delta(S)} c(e)}{\sum_{i\in\mathcal{I}(\mathcal{S})} d(i)} = \min_{S\in\mathcal{S}} \Phi(S)
\end{aligned}
$$

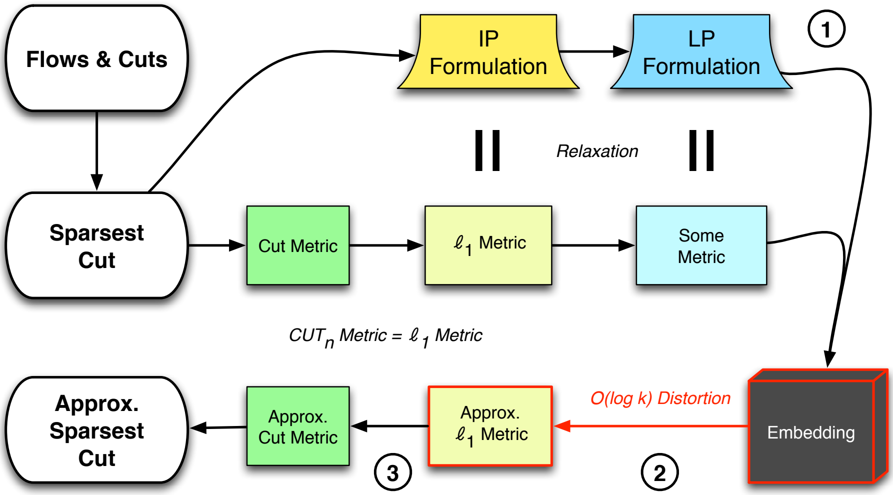
By definition, $\delta_S(u, v) = 1$ only when $e = (u, v)$ crosses the cut defined by S i.e. $e \in \delta(S)$. This means,

$$\begin{aligned}
\frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in E} c(e)\delta_S(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_i d(i)\delta_S(s_i, t_i)} &= \frac{\sum_{S \in \mathcal{S}} \alpha_S \sum_{e \in \delta(S)} c(e)}{\sum_{S \in \mathcal{S}} \alpha_S \sum_{i \in \mathcal{I}(S)} d(i)} \\
&\geq \min_{S \in \mathcal{S}} \frac{\alpha_S \sum_{e \in \delta S} c(e)}{\alpha_S \sum_{i \in \mathcal{I}(S)} d(i)} \\
&= \min_{S \in \mathcal{S}} \frac{\sum_{e \in \delta(S)} c(e)}{\sum_{i \in \mathcal{I}(\mathcal{S})} d(i)} = \min_{S \in \mathcal{S}} \Phi(S) \\
&\Rightarrow \exists S \in \mathcal{S} \text{ s.t. } \Phi(S) \leq \phi
\end{aligned}$$

Roadmap


Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
Embedding
3
2

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
O(log k) Distortion
Embedding
3
2

What are Metric Embeddings?

What are Metric Embeddings?

Definition

Given metric spaces (X, d) and (X', d'), a map $g : X \to X'$ is an *isometric* embedding if, $d(x, y) = d'(g(x), g(y)) \;\forall\; x, y \;\in\; X$

What are Metric Embeddings?

Definition

Given metric spaces (X, d) and (X', d'), a map $g : X \to X'$ is an *isometric* embedding if, $d(x, y) = d'(g(x), g(y)) \; \forall \; x, y \; \in \; X$

Non-isometric embeddings:

What are Metric Embeddings?

Definition

Given metric spaces (X, d) and (X', d'), a map $g : X \to X'$ is an *isometric* embedding if, $d(x, y) = d'(g(x), g(y)) \; \forall \; x, y \; \in \; X$

Non-isometric embeddings:

1. Contraction of g: $\alpha = \max_{x,y \in X} \dfrac{d(x, y)}{d'(g(x), g(y))}$

What are Metric Embeddings?

Definition

Given metric spaces (X, d) and (X', d'), a map $g : X \to X'$ is an *isometric* embedding if, $d(x, y) = d'(g(x), g(y)) \; \forall \; x, y \; \in \; X$

Non-isometric embeddings:

1. Contraction of g: $\alpha = \max\limits_{x,y \in X} \dfrac{d(x, y)}{d'(g(x), g(y))}$
2. Expansion of g: $\beta = \max\limits_{x,y \in X} \dfrac{d'(g(x), g(y))}{d(x, y)}$

What are Metric Embeddings?

Definition

Given metric spaces (X, d) and (X', d'), a map $g : X \to X'$ is an *isometric* embedding if, $d(x, y) = d'(g(x), g(y)) \; \forall \; x, y \; \in \; X$

Non-isometric embeddings:

1. Contraction of g: $\alpha = \max\limits_{x,y \in X} \dfrac{d(x, y)}{d'(g(x), g(y))}$
2. Expansion of g: $\beta = \max\limits_{x,y \in X} \dfrac{d'(g(x), g(y))}{d(x, y)}$
3. Distortion of g: $||g||_{dist} = \alpha \cdot \beta$

Embedding into an ℓ_1 metric

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \dots k\}$, where $|T| = 2k = 2^\tau$

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \dots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v) =$ shortest path distance $u, v \in V$ w.r.t metric x

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \ldots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v)$ = shortest path distance $u, v \in V$ w.r.t metric x
- For $A \subseteq V$ and $u \in V : \text{dist}_x(u, A) = \min_{v \in A} \text{dist}_x(u, v)$

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \ldots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v) =$ shortest path distance $u, v \in V$ w.r.t metric x
- For $A \subseteq V$ and $u \in V$: $\text{dist}_x(u, A) = \min_{v \in A} \text{dist}_x(u, v)$
- For the dimensions of the embedding ($\mathcal{D}$):
  $\mathcal{D} = \tau L$ where, $L = q \log k$; $\tau = \log k$; $\Rightarrow \mathcal{D} = O(\log^2 k)$

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \dots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v)$ = shortest path distance $u, v \in V$ w.r.t metric x
- For $A \subseteq V$ and $u \in V$: $\text{dist}_x(u, A) = \min_{v \in A} \text{dist}_x(u, v)$
- For the dimensions of the embedding ($\mathcal{D}$):
  $\mathcal{D} = \tau L$ where, $L = q \log k$; $\tau = \log k$; $\Rightarrow \mathcal{D} = O(\log^2 k)$

Embedding

Repeat for $l = 1 \dots L, t = 1 \dots \tau$:

Embedding into an ℓ_1 metric

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \ldots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v) =$ shortest path distance $u, v \in V$ w.r.t metric x
- For $A \subseteq V$ and $u \in V$: $\text{dist}_x(u, A) = \min_{v \in A} \text{dist}_x(u, v)$
- For the dimensions of the embedding ($\mathcal{D}$): $\mathcal{D} = \tau L$ where, $L = q \log k$; $\tau = \log k$; $\Rightarrow \mathcal{D} = O(\log^2 k)$

Embedding

Repeat for $l = 1 \ldots L, t = 1 \ldots \tau$:

- Construct sets A_{tl} each of which has $\frac{k}{2^t} = 2^{\tau - t}$ points sampled with replacement from T

- Space of terminal vertices $T = \{s_i, t_i | i = 1 \ldots k\}$, where $|T| = 2k = 2^\tau$
- $\text{dist}_x(u, v) =$ shortest path distance $u, v \in V$ w.r.t metric x
- For $A \subseteq V$ and $u \in V$: $\text{dist}_x(u, A) = \min_{v \in A} \text{dist}_x(u, v)$
- For the dimensions of the embedding ($\mathcal{D}$):
  $\mathcal{D} = \tau L$ where, $L = q \log k$; $\tau = \log k$; $\Rightarrow \mathcal{D} = O(\log^2 k)$

Embedding

Repeat for $l = 1 \ldots L, t = 1 \ldots \tau$:

- Construct sets A_{tl} each of which has $\frac{k}{2^t} = 2^{\tau - t}$ points sampled with replacement from T
- Also, define $f_{tl}(v) = \text{dist}_x(v, A_{tl}) \; \forall \; v \in V$

Is it a Good Embedding?

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (2):

Is it a Good Embedding?

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \dots k$

From (2):

$$\sum_{i=1}^{k} d_i ||f(s_i) - f(t_i)||_1 \quad \geq \quad \sum_{i=1}^{k} d_i y(i) L/88 = \Omega(L \sum_{i=1}^{k} d_i y(i))$$

Is it a Good Embedding?

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (2):

$$\begin{aligned}\sum_{i=1}^{k} d_i ||f(s_i) - f(t_i)||_1 &\geq \sum_{i=1}^{k} d_i y(i) L/88 = \Omega(L \sum_{i=1}^{k} d_i y(i)) \\ &= \Omega(L) \ \ (\because \sum_{i=1}^{k} d_i y(i) = 1)\end{aligned}$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (1):

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (1):

$$\sum_{(u,v)=e\in E} c(e)||f(u) - f(v)||_1 \quad \leq \quad \sum_{(u,v)=e\in E} \mathcal{D}c(e)x(e)$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (1):

$$\begin{aligned} \sum_{(u,v)=e \in E} c(e)||f(u) - f(v)||_1 &\leq \sum_{(u,v)=e \in E} \mathcal{D}c(e)x(e) \\ &= \mathcal{D} \sum_{(u,v)=e \in E} c(e)x(e) \end{aligned}$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

From (1):

$$\begin{aligned}
\sum_{(u,v)=e \in E} c(e)||f(u) - f(v)||_1 &\leq \sum_{(u,v)=e \in E} \mathcal{D}c(e)x(e) \\
&= \mathcal{D} \sum_{(u,v)=e \in E} c(e)x(e) \\
&= O(\log^2 k) \sum_{e \in E} c(e)x(e)
\end{aligned}$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)|| \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

This implies,

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)|| \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

This implies,

$$\Phi = \frac{\sum_{(u,v)=e\in E} c(e)||f(u) - f(v)||_1}{\sum_{i=1}^{k} d_i||f(s_i) - f(t_i)||_1} \leq \frac{O(\log^2 k)\sum_{e\in E} c(e)x(e)}{\Omega(L)}$$

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)|| \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

This implies,

$$\Phi = \frac{\sum_{(u,v)=e\in E} c(e)||f(u) - f(v)||_1}{\sum_{i=1}^{k} d_i||f(s_i) - f(t_i)||_1} \leq \frac{O(\log^2 k)\sum_{e\in E} c(e)x(e)}{\Omega(L)}$$

$$= O(\frac{\log^2 k}{\log k})\sum_{e\in E} c(e)x(e)$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)|| \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \dots k$

This implies,

$$\begin{aligned}
\Phi = \frac{\sum_{(u,v)=e\in E} c(e)||f(u) - f(v)||_1}{\sum_{i=1}^{k} d_i||f(s_i) - f(t_i)||_1} &\leq \frac{O(\log^2 k)\sum_{e\in E} c(e)x(e)}{\Omega(L)} \\
&= O(\frac{\log^2 k}{\log k})\sum_{e\in E} c(e)x(e) \\
&= O(\log k)\sum_{e\in E} c(e)x(e)
\end{aligned}$$

Is it a Good Embedding?

Continued...

Lemmas

Lemma 1: For each edge $e = (u, v)$, $||f(u) - f(v)|| \leq \mathcal{D}x(e)$
Lemma 2: With probability at least $\frac{1}{2}$:
$||f(s_i) - f(t_i)||_1 \geq L \cdot y(i)/88$ for each $i = 1 \ldots k$

This implies,

$$\begin{aligned} \Phi = \frac{\sum_{(u,v)=e\in E} c(e)||f(u) - f(v)||_1}{\sum_{i=1}^{k} d_i||f(s_i) - f(t_i)||_1} &\leq \frac{O(\log^2 k)\sum_{e\in E} c(e)x(e)}{\Omega(L)} \\ &= O(\frac{\log^2 k}{\log k})\sum_{e\in E} c(e)x(e) \\ &= O(\log k)\sum_{e\in E} c(e)x(e) \\ &= O(logk)\ \Phi^* \end{aligned}$$

Proving Lemma 1

Lemma

For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proving Lemma 1

Lemma

For each edge $e = (u, v), ||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proof: For any $A \subseteq V$, $e = (u, v) \in E$,

Proving Lemma 1

Lemma

For each edge $e = (u, v), ||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proof: For any $A \subseteq V,\ e = (u, v)\ \in\ E$,

$$\begin{aligned} \text{dist}_x(u, A) &\leq \text{dist}_x(u, v) + \text{dist}_x(v, A) \\ \text{dist}_x(v, A) &\leq \text{dist}_x(u, v) + \text{dist}_x(v, A) \end{aligned}$$

Proving Lemma 1

Lemma

For each edge $e = (u, v)$, $||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proof: For any $A \subseteq V$, $e = (u, v) \in E$,

$$\begin{aligned}
\mathrm{dist}_x(u, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\mathrm{dist}_x(v, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\Rightarrow \mathrm{dist}_x(u, A) - \mathrm{dist}_x(v, A) &\leq x(e) \ (\because \mathrm{dist}_x(u, v) = x(e); e = (u, v)) \\
\mathrm{dist}_x(v, A) - \mathrm{dist}_x(u, A) &\leq x(e)
\end{aligned}$$

Proving Lemma 1

Lemma

For each edge $e = (u, v), ||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proof: For any $A \subseteq V,\ e = (u, v)\ \in\ E$,

$$
\begin{aligned}
\mathrm{dist}_x(u, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\mathrm{dist}_x(v, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\Rightarrow \mathrm{dist}_x(u, A) - \mathrm{dist}_x(v, A) &\leq x(e)\ (\because \mathrm{dist}_x(u, v) = x(e); e = (u, v)) \\
\mathrm{dist}_x(v, A) - \mathrm{dist}_x(u, A) &\leq x(e) \\
\Rightarrow ||f(u) - f(v)||_1 &= \sum_{t,l} |f_{tl}(u) - f_{tl}(v)|
\end{aligned}
$$

Proving Lemma 1

Lemma

For each edge $e = (u, v), ||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

Proof: For any $A \subseteq V,\ e = (u, v)\ \in\ E$,

$$
\begin{aligned}
\mathrm{dist}_x(u, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\mathrm{dist}_x(v, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\Rightarrow \mathrm{dist}_x(u, A) - \mathrm{dist}_x(v, A) &\leq x(e)\ (\because \mathrm{dist}_x(u, v) = x(e); e = (u, v)) \\
\mathrm{dist}_x(v, A) - \mathrm{dist}_x(u, A) &\leq x(e) \\
\Rightarrow ||f(u) - f(v)||_1 &= \sum_{t,l} |f_{tl}(u) - f_{tl}(v)| \\
&= \sum_{t=1}^{T} \sum_{l=1}^{L} |\,\mathrm{dist}_x(u, A_{tl}) - \mathrm{dist}_x(v, A_{tl})|
\end{aligned}
$$

Proving Lemma 1

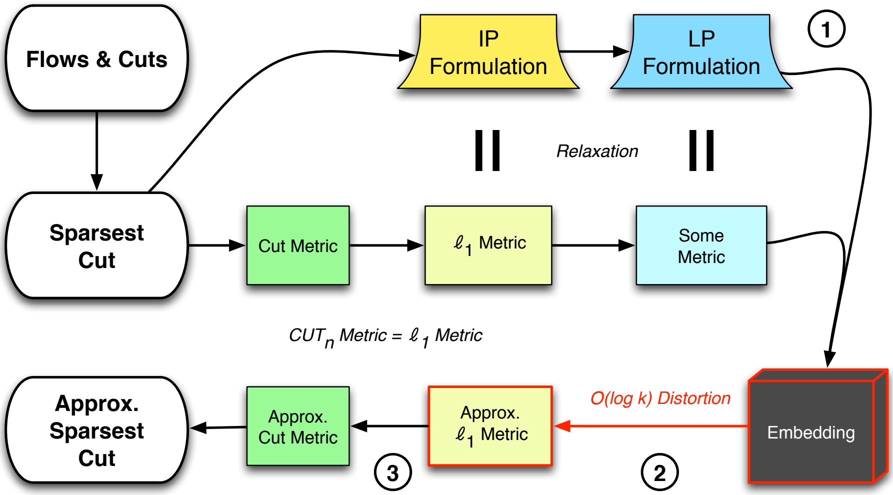
Lemma

For each edge $e = (u, v), ||f(u) - f(v)||_1 \leq \mathcal{D}x(e)$

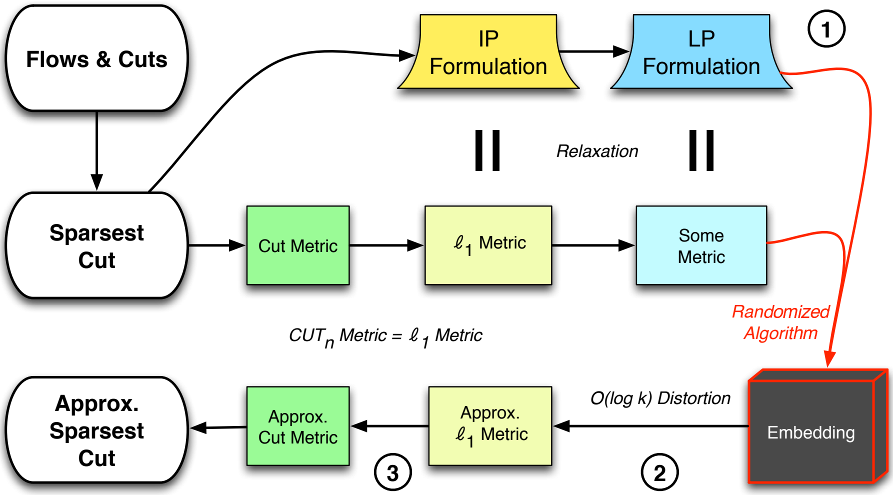
Proof: For any $A \subseteq V,\ e = (u, v)\ \in\ E$,

$$\begin{aligned}
\mathrm{dist}_x(u, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\mathrm{dist}_x(v, A) &\leq \mathrm{dist}_x(u, v) + \mathrm{dist}_x(v, A) \\
\Rightarrow \mathrm{dist}_x(u, A) - \mathrm{dist}_x(v, A) &\leq x(e)\ (\because \mathrm{dist}_x(u, v) = x(e); e = (u, v)) \\
\mathrm{dist}_x(v, A) - \mathrm{dist}_x(u, A) &\leq x(e) \\
\Rightarrow ||f(u) - f(v)||_1 &= \sum_{t,l} |f_{tl}(u) - f_{tl}(v)| \\
&= \sum_{t=1}^{\tau} \sum_{l=1}^{L} |\,\mathrm{dist}_x(u, A_{tl}) - \mathrm{dist}_x(v, A_{tl})| \\
&\leq \tau L x(e) = \mathcal{D}x(e)
\end{aligned}$$

Roadmap


Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
CUT_n Metric = ℓ_1 Metric
O(log k) Distortion
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
Embedding
3
2

Roadmap


Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Randomized Algorithm
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
O(log k) Distortion
Embedding
3
2

Proof of Lemma 2: $||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$, for $i = 1, ..., k$

Lemma 2

$||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$, for $i = 1, ..., k$.

Proof Sketch: We want to

- Concentrate on single (s_i, t_i).
- Show that f embeds s_i, t_i s.t. they are far apart compared to $y(i)$.

Proof of Lemma 2: $||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$, for $i = 1, ..., k$

Lemma 2

$||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$, for $i = 1, ..., k$.

Proof Sketch: We want to

- Concentrate on single (s_i, t_i).
- Show that f embeds s_i, t_i s.t. they are far apart compared to $y(i)$.
- Show each coordinate f_{tl} contributes $(r_t - r_{t-1})$ with high probability.
- By summing over all l, they all would most likely contribute $\Omega(L(r_t - r_{t-1}))$.
- Summing the bound for $t = 1, ..., \hat{t}$, We get sum $= \Omega(Lr_{\hat{t}}) = \Omega(L.y(i))$

Concentrating on single commodity

$T = \{s_i, t_i : i = 1, ..., k\}, |T| = 2k$

For $v \in \{s_i, t_i\}$

$B_x(v, r) = \{w \in T : \text{dist}_x(v, w) \leq r\}$

$B_x^o(v, r) = \{w \in T : \text{dist}_x(v, w) < r\}$

Concentrating on single commodity

$T = \{s_i, t_i : i = 1, ..., k\}, |T| = 2k$
For $v \in \{s_i, t_i\}$
$B_x(v, r) = \{w \in T : \text{dist}_x(v, w) \leq r\}$
$B_x^o(v, r) = \{w \in T : \text{dist}_x(v, w) < r\}$

- Let $r_0 = 0$ and r_t be the smallest r s.t. $|B_x(u, r)| \geq 2^t$, for both $u \in \{s_i, t_i\}$
- Let $\hat{t}$ be the smallest t s.t. $r_{\hat{t}} \geq y(i)/4$,
  Set $r_{\hat{t}} = y(i)/4$
- But $y(i) \leq \text{dist}_x(s_i, t_i)$
- Thus Balls are disjoints.

Proof of Lemma 2: $||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$

Observation:

- $A \cap B_x^0(s_i, r_t) = \emptyset \Leftrightarrow \text{dist}_x(s_i, A) \geq r_t$
- $A \cap B_x(t_i, r_{t-1}) \neq \emptyset \Leftrightarrow \text{dist}_x(t_i, A) \leq r_{t-1}$

Proof of Lemma 2: $||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$

Observation:

- $A \cap B_x^0(s_i, r_t) = \emptyset \Leftrightarrow \text{dist}_x(s_i, A) \geq r_t$
- $A \cap B_x(t_i, r_{t-1}) \neq \emptyset \Leftrightarrow \text{dist}_x(t_i, A) \leq r_{t-1}$
- Let, $E_{tl}, t = 1, ..., \hat{t}, l = 1, ..., L$ denote the event that

Proof of Lemma 2: $||f(s_i) - f(t_i)||_1 \geq L.y(i)/88$

Observation:

- $A \cap B_x^0(s_i, r_t) = \emptyset \Leftrightarrow \text{dist}_x(s_i, A) \geq r_t$
- $A \cap B_x(t_i, r_{t-1}) \neq \emptyset \Leftrightarrow \text{dist}_x(t_i, A) \leq r_{t-1}$
- Let, $E_{tl}, t = 1, ..., \hat{t}, l = 1, ..., L$ denote the event that $A_{tl} \cap B_x^o(s_i, r_t) = \emptyset$ and $A_{tl} \cap B_x(t_i, r_{t-1}) \neq \emptyset$
- E_{tl} implies $|f_{tl}(s_i) - f_{tl}(t_i)| = |\,\text{dist}_x(s_i, A_{tl}) - \text{dist}_x(t_i, A_{tl})| \geq (r_t - r_{t-1})$
- We will show that E_{tl} is likely to occur

Facts of Probability

- Let $G, B \subseteq X$
- A is formed by selecting p elements of X independently, uniformly at random
- $\Pr[A \cap G \neq \emptyset \text{ and } A \cap B = \emptyset]$
  $= \Pr[A \cap G \neq \emptyset | A \cap B = \emptyset].\Pr[A \cap B = \emptyset]$
  $\geq \Pr[A \cap G \neq \emptyset].\Pr[A \cap B = \emptyset]$

Facts of Probability

- Let $G, B \subseteq X$
- A is formed by selecting p elements of X independently, uniformly at random
- $\Pr[A \cap G \neq \emptyset \text{ and } A \cap B = \emptyset]$
  $= \Pr[A \cap G \neq \emptyset | A \cap B = \emptyset].\Pr[A \cap B = \emptyset]$
  $\geq \Pr[A \cap G \neq \emptyset].\Pr[A \cap B = \emptyset]$
- $\forall Y \subseteq X$, $\Pr[A \cap Y = \emptyset] = (1 - \frac{|Y|}{|X|})^p$

Facts of Probability

- Let $G, B \subseteq X$
- A is formed by selecting p elements of X independently, uniformly at random
- $\Pr[A \cap G \neq \emptyset \text{ and } A \cap B = \emptyset]$
  $= \Pr[A \cap G \neq \emptyset | A \cap B = \emptyset].\Pr[A \cap B = \emptyset]$
  $\geq \Pr[A \cap G \neq \emptyset].\Pr[A \cap B = \emptyset]$
- $\forall Y \subseteq X$, $\Pr[A \cap Y = \emptyset] = (1 - \frac{|Y|}{|X|})^p$
- If $p = \frac{|Y|}{|X|}$, and tends to infinity
  $(1 - \frac{|Y|}{|X|})^p$ approaches $1/e$ and always in the interval $[\frac{1}{4}, \frac{1}{e}]$

Facts of Probability

- Let $G, B \subseteq X$
- A is formed by selecting p elements of X independently, uniformly at random
- $\Pr[A \cap G \neq \emptyset \text{ and } A \cap B = \emptyset]$
  $= \Pr[A \cap G \neq \emptyset | A \cap B = \emptyset].\Pr[A \cap B = \emptyset]$
  $\geq \Pr[A \cap G \neq \emptyset].\Pr[A \cap B = \emptyset]$
- $\forall Y \subseteq X$, $\Pr[A \cap Y = \emptyset] = (1 - \frac{|Y|}{|X|})^p$
- If $p = \frac{|Y|}{|X|}$, and tends to infinity
  $(1 - \frac{|Y|}{|X|})^p$ approaches $1/e$ and always in the interval $[\frac{1}{4}, \frac{1}{e}]$
- If $p = \beta\frac{|Y|}{|X|}$ interval is $[(\frac{1}{4})^\beta, (\frac{1}{e})^\beta]$

Coming back to the proof

Coming back to the proof

- $A = A_{tl}$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$
- $B = B_x^o(s_i, r_t), |B| < 2^t$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$
- $B = B_x^o(s_i, r_t), |B| < 2^t$
- $G = B_x(t_i, r_{t-1}), |G| \geq 2^{t-1}$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$
- $B = B_x^o(s_i, r_t), |B| < 2^t$
- $G = B_x(t_i, r_{t-1}), |G| \geq 2^{t-1}$
- $p = 2^{\tau - t}$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$
- $B = B_x^o(s_i, r_t), |B| < 2^t$
- $G = B_x(t_i, r_{t-1}), |G| \geq 2^{t-1}$
- $p = 2^{\tau - t}$
- Hence, $p < \frac{|X|}{|B|}$ and $p \geq \frac{1}{2}\frac{|X|}{|G|}$
  $\rightarrow \Pr[A \cap B = \emptyset] \geq \frac{1}{4}$
  $\rightarrow \Pr[A \cap G \neq \emptyset] \geq (1 - (\frac{1}{e})^{\frac{1}{2}})$

Coming back to the proof

- $A = A_{tl}$
- $X = T, |X| = 2^{\tau}$
- $B = B_x^o(s_i, r_t), |B| < 2^t$
- $G = B_x(t_i, r_{t-1}), |G| \geq 2^{t-1}$
- $p = 2^{\tau - t}$
- Hence, $p < \frac{|X|}{|B|}$ and $p \geq \frac{1}{2}\frac{|X|}{|G|}$
  $\rightarrow \Pr[A \cap B = \emptyset] \geq \frac{1}{4}$
  $\rightarrow \Pr[A \cap G \neq \emptyset] \geq (1 - (\frac{1}{e})^{\frac{1}{2}})$
- $\Pr[E_{tl}] \geq \frac{(1-(\frac{1}{e})^{\frac{1}{2}})}{4} \geq \frac{1}{11}$, for $t = 1, ..., \hat{t}, l = 1, ..., L$

Using Chernoff Bound to Summarize

- If we fix a particular $t = 1, ..., \hat{t}$
  define indicator variable, $X_l \in \{0, 1\}$ for $l = 1, ..., L$
  $X_l = 1 \rightarrow E_{tl}$ occurs

We use Chernoff bound to show that $\sum_{l=1}^{L} x_l$ does not deviate too much from its expectation $E[x_l] \geq \frac{L}{11}$

Using Chernoff Bound to Summarize

- If we fix a particular $t = 1, ..., \hat{t}$
  define indicator variable, $X_l \in \{0, 1\}$ for $l = 1, ..., L$
  $X_l = 1 \rightarrow E_{tl}$ occurs

We use Chernoff bound to show that $\sum_{l=1}^{L} x_l$ does not deviate too much from its expectation $E[x_l] \geq \frac{L}{11}$

Chernoff Bound

If $E[x_l] = \mu$ then $\Pr[\sum_{l=1}^{L} x_l < \frac{\mu}{2}] \leq \exp(-\mu/8)$

Using Chernoff Bound to Summarize

- If we fix a particular $t = 1, ..., \hat{t}$
  define indicator variable, $X_l \in \{0, 1\}$ for $l = 1, ..., L$
  $X_l = 1 \rightarrow E_{tl}$ occurs

We use Chernoff bound to show that $\sum_{l=1}^{L} x_l$ does not deviate too much from its expectation $E[x_l] \geq \frac{L}{11}$

Chernoff Bound

If $E[x_l] = \mu$ then $\Pr[\sum_{l=1}^{L} x_l < \frac{\mu}{2}] \leq \exp(-\mu/8)$

- Since $\mu \geq \frac{L}{11} = \frac{q \log k}{11}$, if say $q = 200 \rightarrow$ Probability is at most $\frac{1}{2k \log 2k}$.

Using Chernoff Bound to Summarize

- If we fix a particular $t = 1, ..., \hat{t}$
  define indicator variable, $X_l \in \{0, 1\}$ for $l = 1, ..., L$
  $X_l = 1 \rightarrow E_{tl}$ occurs

We use Chernoff bound to show that $\sum_{l=1}^{L} x_l$ does not deviate too much from its expectation $E[x_l] \geq \frac{L}{11}$

Chernoff Bound

If $E[x_l] = \mu$ then $\Pr[\sum_{l=1}^{L} x_l < \frac{\mu}{2}] \leq \exp(-\mu/8)$

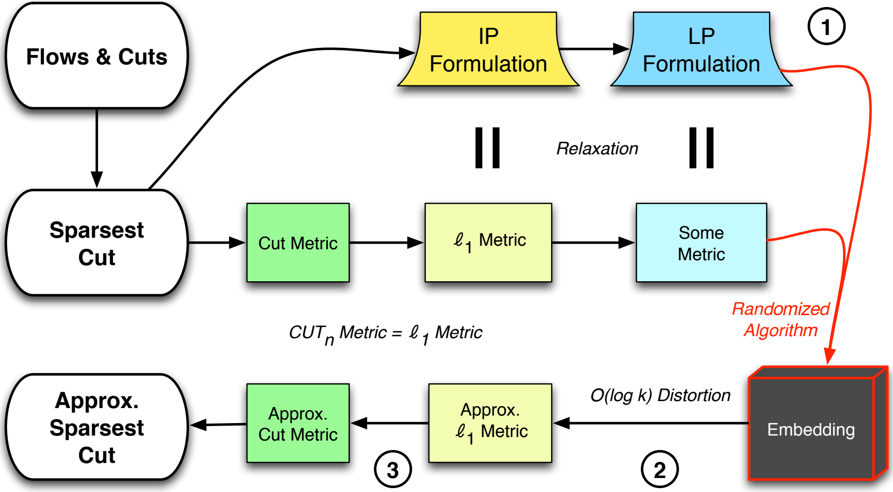
- Since $\mu \geq \frac{L}{11} = \frac{q \log k}{11}$, if say $q = 200 \rightarrow$ Probability is at most $\frac{1}{2k \log 2k}$.
- Most importantly, if $\sum_l x_l \geq \frac{L}{22}$ then we know that for $\frac{L}{22}$ of the components $f_{tl}, l = 1, ..., L$ E_{tl} occurs
- so, $\sum_{l=1}^{L} |f_{tl}(s_i) - f_{tl}(t_i)| \geq (r_t - r_{t-1})\frac{L}{22}$

Final pieces of the proof

- $\sum_{l=1}^{L} |f_{tl}(s_i) - f_{tl}(t_i)| \geq (r_t - r_{t-1})\frac{L}{22}$
- We showed that for any fixed value of $t = 1, .., \hat{t}$, above fails to hold with probability less than $\frac{1}{2k \log 2k}$.
- Since $\hat{t} < \log(2k)$, the above holds for every $t = 1, .., \hat{t}$ with probability at least $1 - \frac{1}{2k}$.
- Hence, with Probability $\geq 1 - \frac{1}{2k}$,
  $\sum_{t=1}^{\hat{t}} \sum_{l=1}^{L} |f_{tl}(s_i) - f_{tl}(t_i)| \geq \sum_{t=1}^{\hat{t}}(r_t - r_{t-1})\frac{L}{22} = r_{\hat{t}}\frac{L}{22} = y(i)\frac{L}{88}$

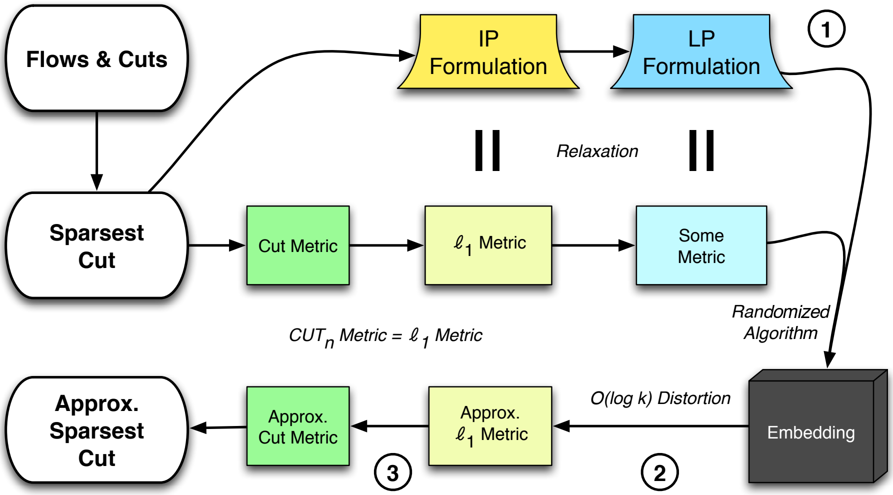
Finally, we can conclude that using the above results and union bounds, **Lemma** 2 **holds for all** i **with high probability**.

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Randomized Algorithm
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
O(log k) Distortion
Embedding
3
2

Roadmap



Flows & Cuts
IP Formulation
LP Formulation
1
Relaxation
Sparsest Cut
Cut Metric
ℓ_1 Metric
Some Metric
Randomized Algorithm
CUT_n Metric = ℓ_1 Metric
Approx. Sparsest Cut
Approx. Cut Metric
Approx. ℓ_1 Metric
O(log k) Distortion
Embedding
3
2

References

- [Vaz01] V. Vazirani (2001), Sparsest Cut, Chapter 21. *Approximation Algorithms*, Springer-Verlag, New York
- [LLR95] Nathan Linial and Eran London and Yuri Rabinovich (1995),The geometry of graphs and some of its algorithmic applications. *Combinatorica*, Vol 15, 577–591
- [CMU19] Anupam Gupta (2008),Lecture 19, Sparsest Cuts and L_1 Embeddings. *CMU Lecture Notes*.
- [ROY07] Roy Liu (2007), Multicommodity Flows Handout
- [Shm97] David B. Shmoys (1997), Cut Problems and their applications to divide-and-conquer. *Approximation Algorithms for NP-hard Problems*, 192–235
- [Mat90] Matula and Shahrokhi (1990), Sparsest Cuts and bottlenecks on graphs, *Discrete Applied Math*, 113–123
- [ALN05] Arora, Lee and Naor (2005), Euclidean distortion and the sparsest cut, *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 553–562
- [LR99] Tom Leighton and Satish Rao (1999), Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, *J. ACM*, 787–832