SPARSEST CUT AND METRIC EMBEDDING

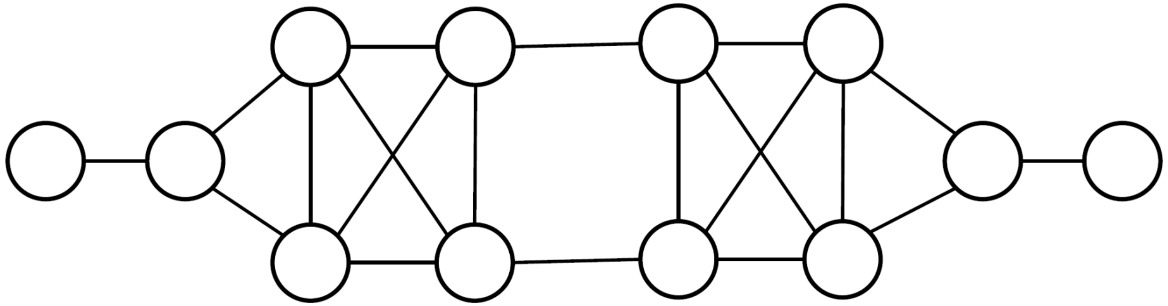
Based on Potechin's PPT

Xue Chen

USTC 2025 Spring

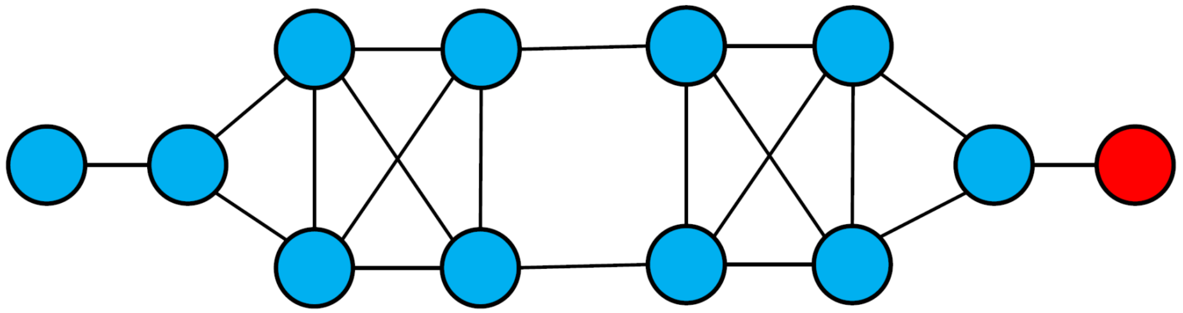
INTRODUCTION

- While Min-Cut admits efficient algorithms (e.g., Karger's algorithm, max-flow min-cut THM),
- it may NOT be the best way to decompose *G*
- Example:

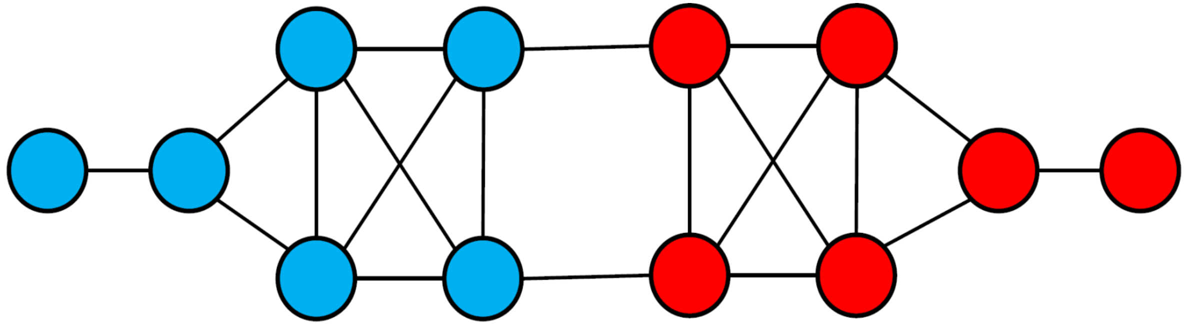


ISSUE OF MIN-CUT

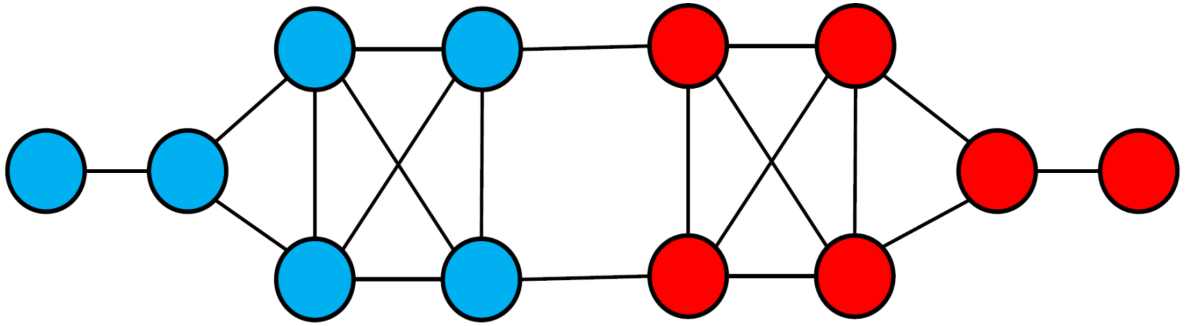
- Min-Cut



- Desired Cut



SPARSEST CUT

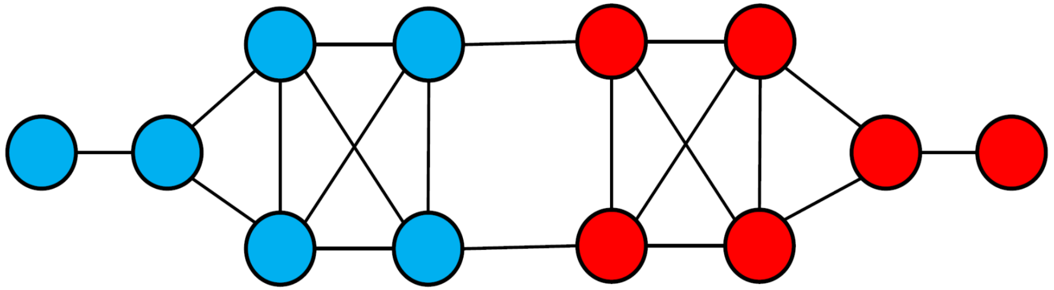


- Intuition: Divide # edges by the size of the smaller side --- $\frac{|E(S,\bar{S})|}{\min\{|S|,|\bar{S}|\}}$
- NP-hard to optimize it (even for any constant approximation)
- However, $\min\{|S|, |\bar{S}|\}$ is not a smooth function

Equivalent up to a factor 2

- Definition: For a cut $S \subset V$, define $\phi(S) = \frac{|E(S,\bar{S})|}{|S|\cdot|\bar{S}|}$
- Goal: Find S to minimize $\phi(S)$

EXTENSION 1



- In many cases, we are more interested in *balanced*-cut like $c_\beta = \min_{|S|,|\bar{S}| \geq \beta n} |E(S, \bar{S})|$ for $\beta = 0.01$

  --- equivalent to sparsest cut essentially

- Claim: If $\exists$efficient ρ-approx. algorithms for sparsest cut, then $\exists$ algorithms finds $\frac{\beta}{2}$-balanced cut with capacity $\leq \frac{\rho}{\frac{\beta}{2}(1-\frac{\beta}{2})} \cdot c_\beta$.

- Analysis:

1. Sparsest cut $\phi \leq \frac{C_\beta}{\beta(1-\beta)n^2}$ in G
2. $E(S_1, \bar{S_1}) \leq n \cdot |S_1| \cdot \phi = \frac{\rho \cdot C_\beta}{\beta(1-\beta)n} \cdot |S_1|$
3. Done if $|S_1| \geq \frac{\beta}{2} n$ otherwise consider $G \setminus S_1$
4. OBS: $\phi' \leq \frac{C_\beta}{(\beta n - |S_1|)(n - |S_1|)}$

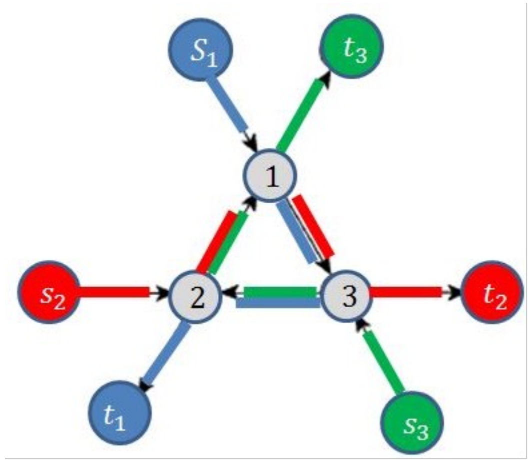
Algorithm:

1. $i = 0$
2. While $|S_1 \cup \cdots \cup S_i| \leq \frac{\beta}{2} n$
   1. $i = i + 1$
   2. Apply ρ-approx. sparsest-cut algorithms to find S_i in the residue $G \setminus (S_1 \cup \cdots \cup S_i)$
3. Output $S_1 \cup \cdots \cup S_i$

EXTENSION 2

Given edge-weights c_e and demands $D_{i,j}$, minimize $\Phi(S) = \frac{\sum_{e \in E(S,\bar{S})} c_e}{\sum_{(i,j) \in (S,\bar{S}) \cup (\bar{S},S)} D_{i,j}}$

- Extension of Max-flow: called multi-commodity flow
- Lots of applications in scheduling, transportation, communication



BACK TO SPARSEST CUT ALGORITHMS

- THM 1 [Leighton-Rao 99]: $\exists$ a linear program relaxation for sparsest cut and a rounding algorithm whose approximation is $O(\log n)$.
- THM 2 [Arora-Rao-Vazirani 07]: $\exists$ a semidefinite program relaxation and a rounding algorithm whose approximation is $O(\sqrt{\log n})$.
- Road map of THM 1:

J. Bourgain:
Fields Medal 1994

Metric spaces: Linear Program $\xrightarrow{\text{[Bourgain 85]}}$ L_1 spaces: Loss factor $O(\log n)$ $\longleftrightarrow$ Cut spaces: Find the cut $(S, \bar{S})$

L_2^2 spaces: Semidefinite Program $\xrightarrow{\text{THM 2 [Arora-Rao-Vazirani 07]}}$ Cut spaces: Loss factor $O(\sqrt{\log n})$

METRIC SPACES

$d + d'$, $\mathrm{d} + 2\mathrm{d}'$ are valid distances

- Definition: A metric space (X, d) is a set of points X and a distance function $d: X \times X \to \mathbf{R}_{\geq 0}$ such that
  1. $\forall x_1, x_2, d(x_1, x_2) = d(x_2, x_1)$
  2. $\forall x_1, x_2, d(x_1, x_2) = 0$ iff $x_1 = x_2$
  3. Triangle-inequality: $d(x_1, x_2) \leq d(x_1, y) + d(y, x_2)\ \forall y \in X$
- Example 1: Euclidean space in $\mathbf{R}^{\boldsymbol{n}}$: $d(x, y) = \|x - y\| \coloneqq \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Example 2: L1 space in $\mathbf{R}^{\boldsymbol{n}}$: $\|x - y\|_1 \coloneqq \sum_{i=1}^{n}|x_i - y_i|$
- Pseudo-metric space: Remove condition 2

EXAMPLE3: CUT SPACES

- Given G, any cut $(S, \bar{S})$ provides a pseudo-metric space:
  1. $d_S(u, v) = 1$ iff u and v are not on the same side;
  2. otherwise $d_S(u, v) = 0$ if both are in S or $\bar{S}$
- All these functions d_S constitute the cut spaces
- Sparsest cut $\frac{|E(S,\bar{S})|}{|S|\cdot|\bar{S}|}$ becomes $\min_d \frac{\sum_{(i,j)\in E} d(i,j)}{\sum_{i<j} d(i,j)}$ over all cut spaces

NEW GOAL

- Consider optimizing $\min_{d} \frac{\sum_{(i,j)\in E} d(i,j)}{\sum_{i<j} d(i,j)}$ over all cut spaces
- 2 issues:

1. Objective function is non-linear
   --- add a constraint $\sum_{i<j} d(i,j) = 1$
2. Cut spaces are difficult to describe
   --- relax cut spaces to **all pseudo-metrices**

- **Key point:** constraints in pseudo-metrices are linear

Linear Program:

$$\min \sum_{i<j:(i,j)\in E} d(i,j)$$

s.t. $\sum_{i<j} d(i,j) = 1$

$d(i,j) = d(j,i)\ \forall i,j$

$d(i,j) \le d(i,k) + d(k,i)\ \forall i,j,k$

METRIC EMBEDDING

Linear Program:

$$\min \sum_{i<j:(i,j)\in E} d(i,j)$$

s.t. $\sum_{i<j} d(i,j) = 1$

$d(i,j) = d(j,i)\ \forall i,j$

$d(i,j) \le d(i,k) + d(k,i)\ \forall i,j,k$

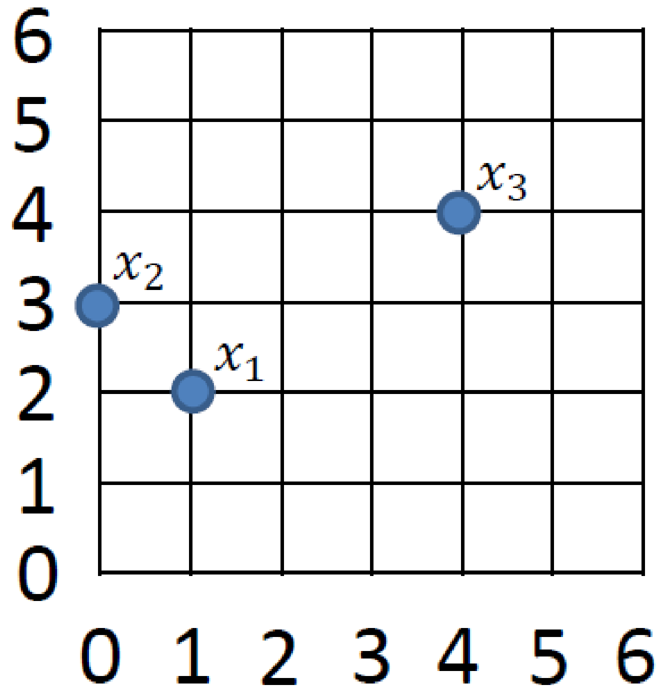
- While this LP is a valid relaxation, how to round it?
- Theorem [Bourgain 85]: For any metric d on n points, $\exists$ a metric $\mu \in L_1$ such that

$$d(\mathrm{x},y) \le \mu(x,y) \le O(\log n)\, d(x,y) \qquad \forall x \neq y$$

most technical part, next

- Fact: Any $\mu \in L_1$ is in cut spaces.
- More importantly, both parts have efficient algorithms

L_1 SPACES



- Example: $d(x_1, x_2) = 2, d(x_1, x_3) = 5, d(x_2, x_3) = 5$
- Lemma 1: Any L_1 Space is a linear combination of cut spaces
- Lemma 2: For any $a, b, c, d \geq 0, \min\left\{\frac{a}{b}, \frac{c}{d}\right\} \leq \frac{a+c}{b+d} \leq max\left\{\frac{a}{b}, \frac{c}{d}\right\}$
- These together implies: If there is a good L_1 embedding, we will find a good sparsest cut!

LAST PIECE: METRIC EMBEDDING INTO L_1

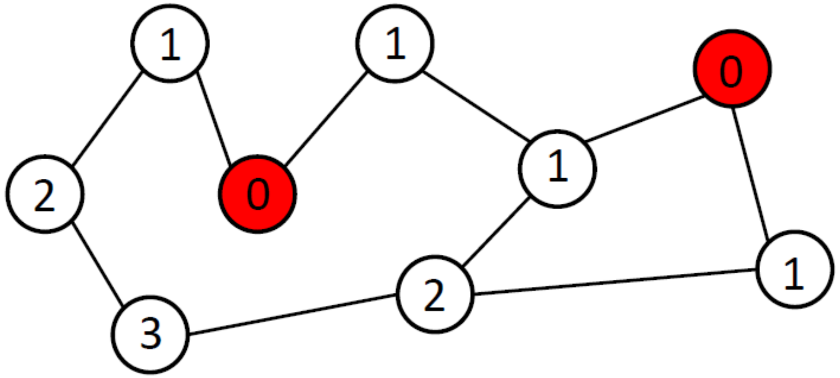
- Theorem [Bourgain 85]: For any metric d on n points, $\exists$ a metric $\mu \in L_1$ such that

$$d(\mathrm{x}, y) \leq \mu(x, y) \leq O(\log n) \cdot d(x, y) \qquad \forall x \neq y$$

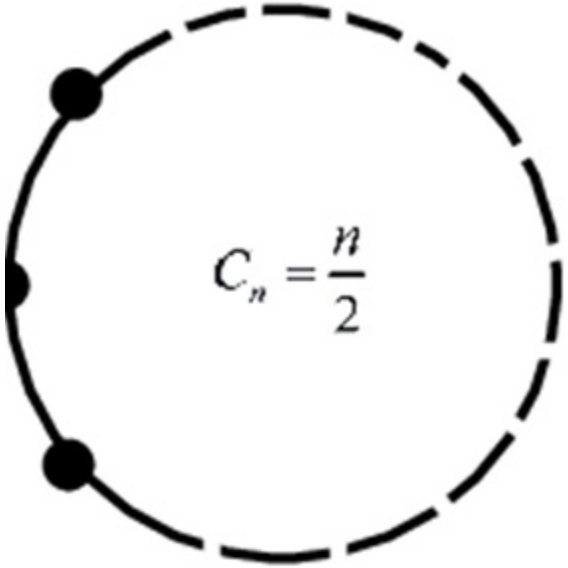
Proof:

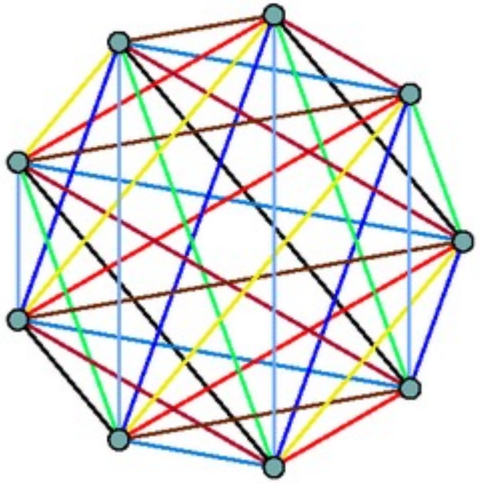
1. Based on Fréchet embedding: let $d(x, S) := \min_{s \in S} d(x, s)$ and $d_S(x, y) := |d(x, S) - d(y, S)|$ as the new distance induced by S
2. Intuition: generate L_1 embedding by combining $d_S(x, y)$ for lots of subsets S

Frećhet EMBEDDING



- Recall $d(x,S) := \min_{s \in S} d(x,s)$ and $d_S(x,y) := |d(x,S) - d(y,S)|$
- Fact: $d_S(x,y) \le d(x,y)$
- 2 extreme examples:

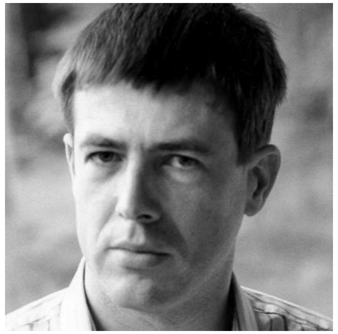




K_n

SUMMARY OF SPARSEST

- $O(\log n)$-approximation based on Linear Program
- $O(\sqrt{\log n})$-approximation based on Semidefinite Program
- Key idea: optimizing over metric spaces then embedding
- Tool: Bourgain's powerful embedding



J. Bourgain:
菲尔兹奖（1994）