AVERAGE SENSITIVITY OF GRAPH ALGORITHMS

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

Nithin Varma
Joint work with Yuichi Yoshida

Sensitivity of an Algorithm

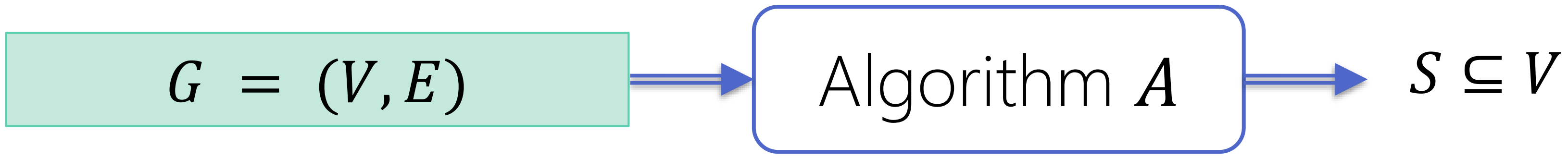
- Measure of change in output as a function of change in input

This talk: A sensitivity definition for graph algorithms

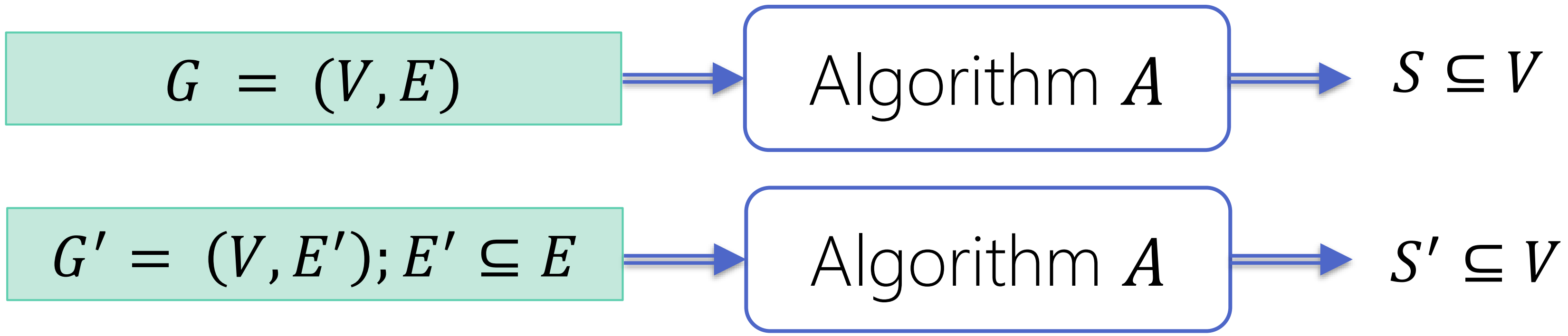
Talk Outline

- Our definition of sensitivity for graph algorithms
- Key properties of our definition
- Main results
- Algorithm with low sensitivity for the global minimum cut problem
- Conclusions and open directions

Average Sensitivity: Intuitive Definition

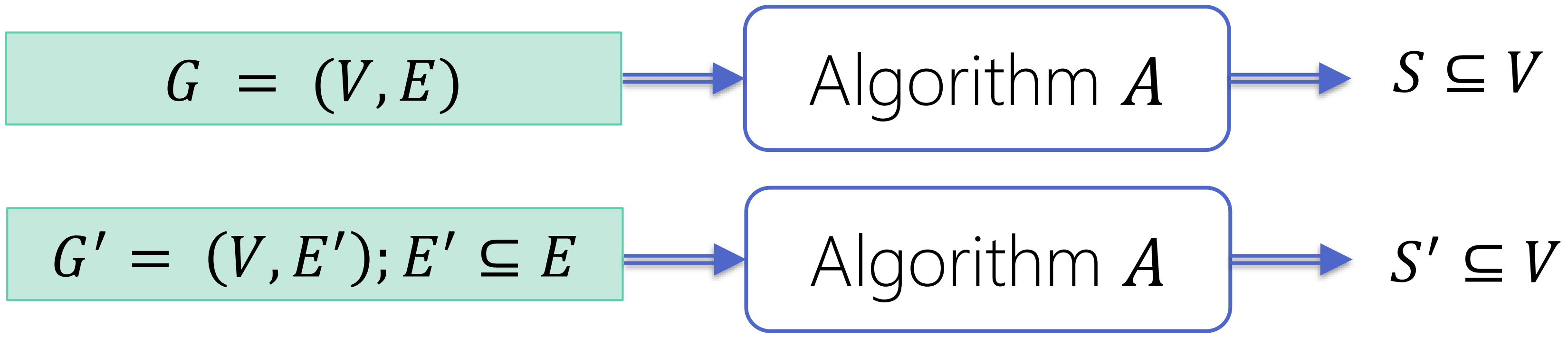


Average Sensitivity: Intuitive Definition



G' is a large subgraph of G obtained by removing a few **random edges**

Average Sensitivity: Intuitive Definition



G' is a large subgraph of G obtained by removing a few **random edges**

Sensitivity of A on $G = |S \,\Delta\, S'| = \mathrm{Ham}(S, S')$

Why Sensitivity?

- Natural notion of performance of algorithms

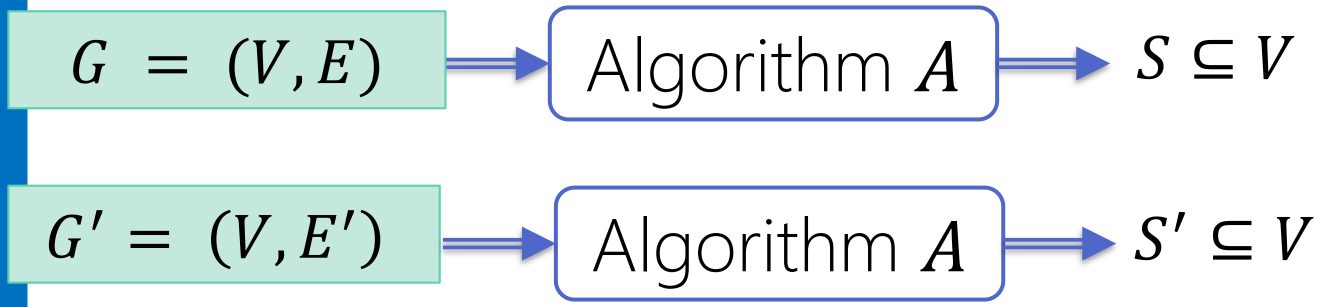
$G = (V, E)$ → Algorithm A → $S \subseteq V$

$G' = (V, E')$ → Algorithm A → $S' \subseteq V$

G' is a large subgraph of G obtained by removing a few random edges

Sensitivity of A $= |S \, \Delta \, S'|$ = Ham(S, S')

Why Sensitivity?



- Natural notion of performance of algorithms
- Answer questions about G by answering questions about G'
  - Useful in cases where one has access only to G'

G' is a large subgraph of G obtained by removing a few random edges

Sensitivity of A $= |S \,\Delta\, S'|$ = Ham(S, S')

Average Sensitivity: Deterministic Algorithms

Deterministic graph algorithm A outputs a set of edges or vertices

Average Sensitivity: Deterministic Algorithms

Deterministic graph algorithm A outputs a set of edges or vertices

Average sensitivity of A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E} \left[\mathrm{Ham}\big(A(G), A(G - e)\big)\right]$$

Average Sensitivity: Deterministic Algorithms

Deterministic graph algorithm A outputs a set of edges or vertices

Average sensitivity of A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E} \left[\mathrm{Ham}\big(A(G), A(G-e)\big)\right]$$

Algorithm with low average sensitivity: **stable-on-average algorithm**

Average Sensitivity: Deterministic Algorithms

Deterministic graph algorithm $\boldsymbol{A}$ outputs a set of edges or vertices

Average sensitivity of $\boldsymbol{A}$ on graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$

$$\text{avg}_{e \in E} \left[\text{Ham}\big(A(G), A(G - e)\big)\right]$$

Algorithm with low average sensitivity: **stable-on-average algorithm**

Generalization to $\boldsymbol{k}$-average sensitivity for the removal of $\boldsymbol{k}$ random edges (without replacement)

Average Sensitivity: Deterministic Algorithms

- **Averaging over edges**: Models random edge deletion from input graphs

Deterministic graph algorithm A outputs a set of edges or vertices

Average sensitivity of A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E} \left[\mathrm{Ham}\big(A(G), A(G - e)\big)\right]$$

Average Sensitivity: Deterministic Algorithms

- **Averaging over edges**: Models random edge deletion from input graphs
- **Sensitivity of solutions, not values**: Solutions may be used in further processing

Deterministic graph algorithm A outputs a set of edges or vertices

Average sensitivity of A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E} \left[\mathrm{Ham}\big(A(G), A(G - e)\big)\right]$$

Example 1: Average Sensitivity of Outputting Large Degree Vertices

Large Degree Vertices

On input G of n vertices:
- Output all vertices of degree at least $n/2$.

Example 1: Average Sensitivity of Outputting Large Degree Vertices

Large Degree Vertices

On input G of n vertices:
- Output all vertices of degree at least $n/2$.

Removing any edge affects the degrees of at most 2 vertices

Example 1: Average Sensitivity of Outputting Large Degree Vertices

Large Degree Vertices

On input G of n vertices:

- Output all vertices of degree at least $n/2$.

Removing any edge affects the degrees of at most 2 vertices

Average sensitivity at most 2

Example 2: Average Sensitivity of s-t Shortest Path

Problem: Given a graph G on n vertices and two vertices s, t, output the s-t shortest path

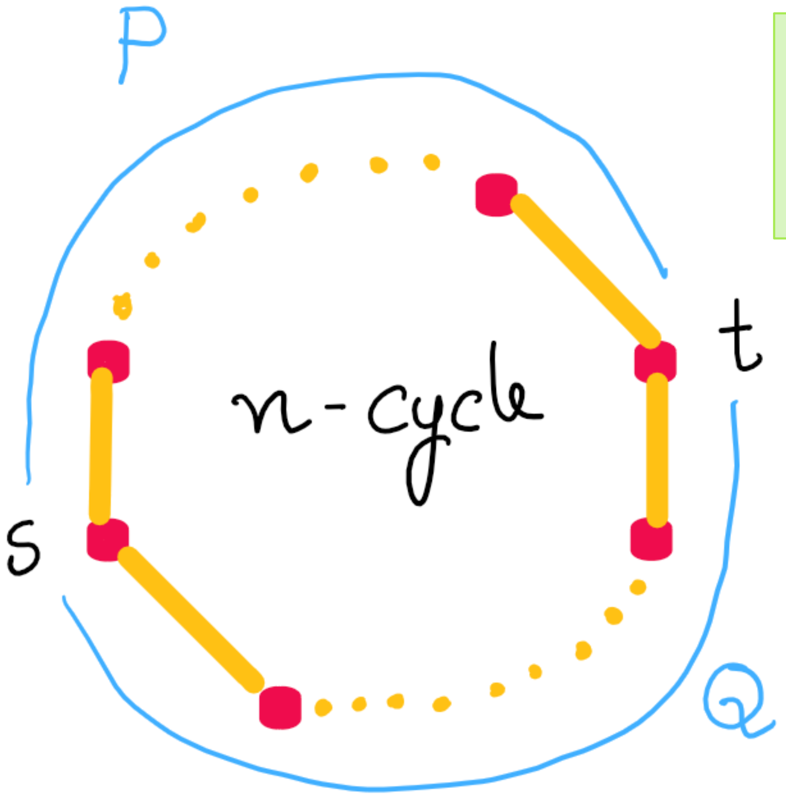
Example 2: Average Sensitivity of s-t Shortest Path

Problem: Given a graph $\boldsymbol{G}$ on $\boldsymbol{n}$ vertices and two vertices $\boldsymbol{s}, \boldsymbol{t}$, output the $\boldsymbol{s}$-$\boldsymbol{t}$ shortest path

Average sensitivity of outputting $\boldsymbol{s}$-$\boldsymbol{t}$ shortest paths is $\Theta(n)$

Example 2: Average Sensitivity of s-t Shortest Path

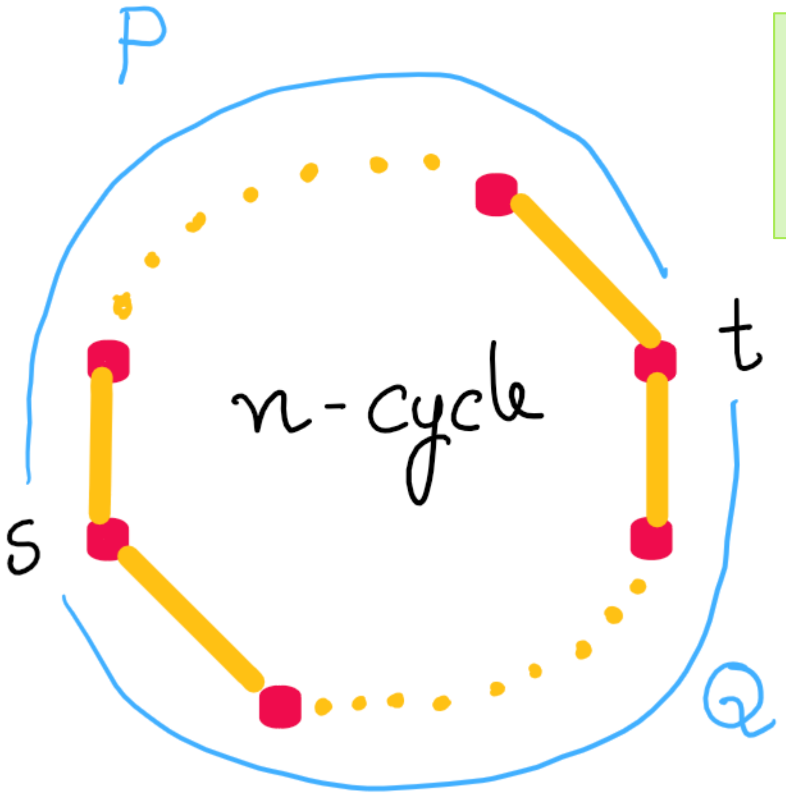
Problem: Given a graph G on n vertices and two vertices s, t, output the s-t shortest path



Average sensitivity of outputting s-t shortest paths is $\Theta(n)$

Example 2: Average Sensitivity of s-t Shortest Path

Problem: Given a graph G on n vertices and two vertices s, t, output the s-t shortest path

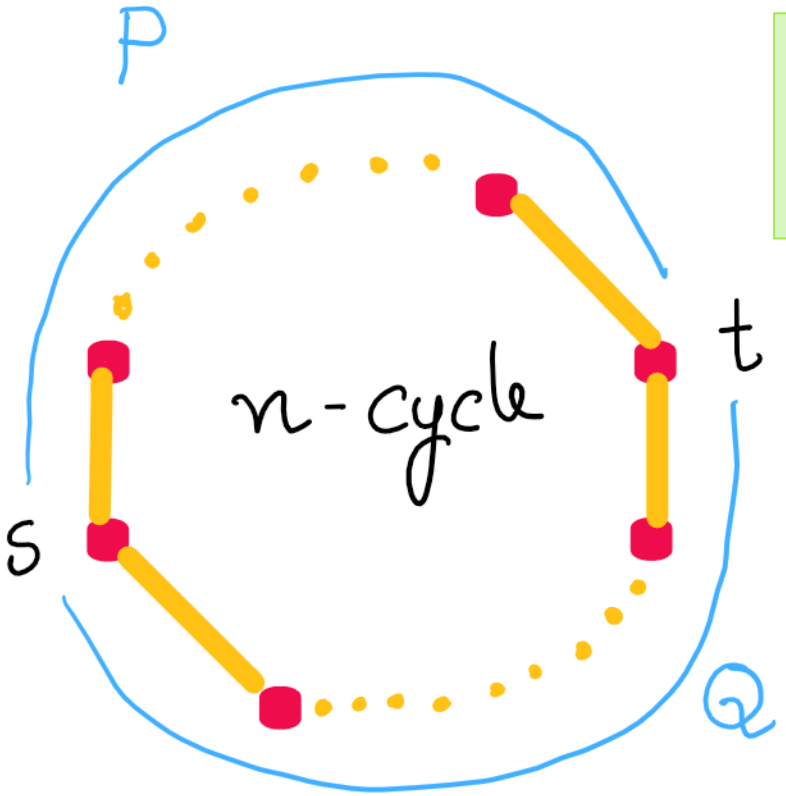


Average sensitivity of outputting s-t shortest paths is $\Theta(n)$

Lower Bound: Consider a deterministic algorithm that outputs P

Example 2: Average Sensitivity of s-t Shortest Path

Problem: Given a graph G on n vertices and two vertices s, t, output the s-t shortest path



Average sensitivity of outputting s-t shortest paths is $\Theta(n)$

For any of the $n/2$ edges removed from P, the algorithm has to output Q

Average Sensitivity: Randomized Algorithms

Distribution over solutions

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E}\left[\mathrm{Dist}\big(A(G), A(G-e)\big)\right]$$

Average Sensitivity: Randomized Algorithms

Distribution over solutions

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E} \left[\mathrm{Dist}\big(A(G), A(G - e)\big)\right]$$

- Earth Mover's Distance
  - Generalization of L_1 distance that penalizes ``significant differences" in probabilities on ``really different" solutions

Average Sensitivity: Randomized Algorithms

Distribution over solutions

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

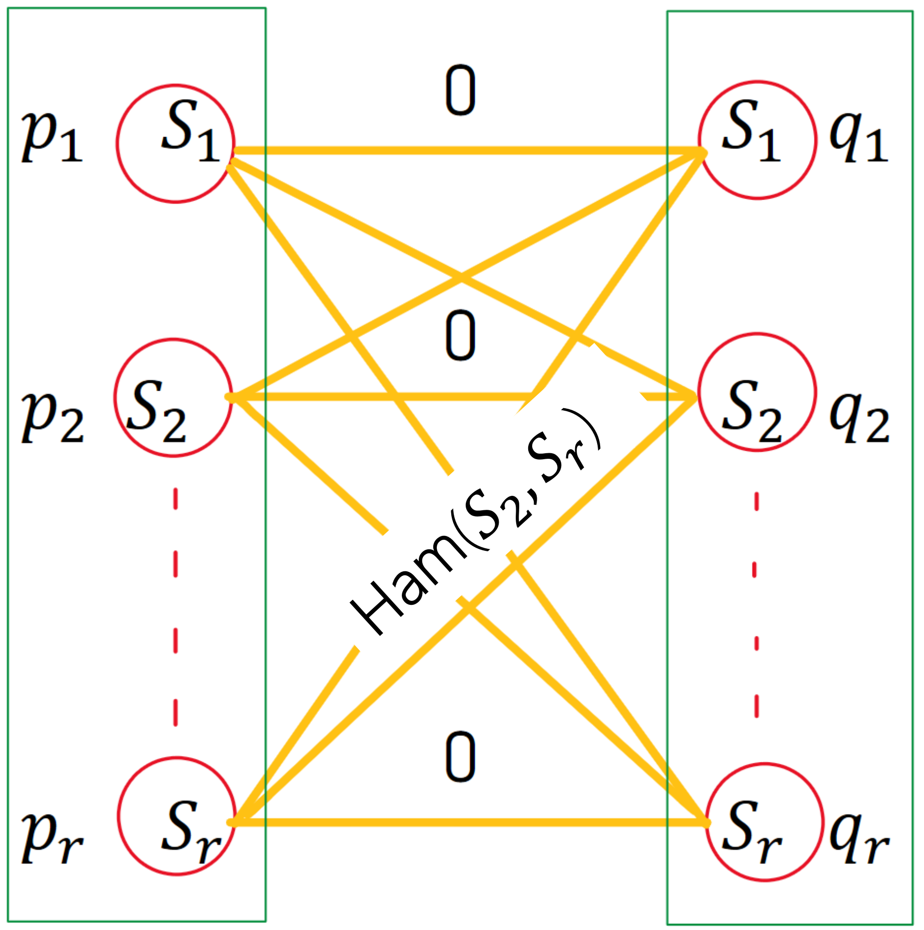
$$\mathrm{avg}_{e \in E}\left[\mathrm{d}_{\mathrm{EM}}\big(A(G), A(G-e)\big)\right]$$

- Earth Mover's Distance
  - Generalization of L_1 distance that penalizes ``significant differences" in probabilities on ``really different" solutions

Average Sensitivity: Randomized Algorithms

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$\text{avg}_{e \in E} \left[\text{d}_{\text{EM}}\left(A(G), A(G-e)\right) \right]$



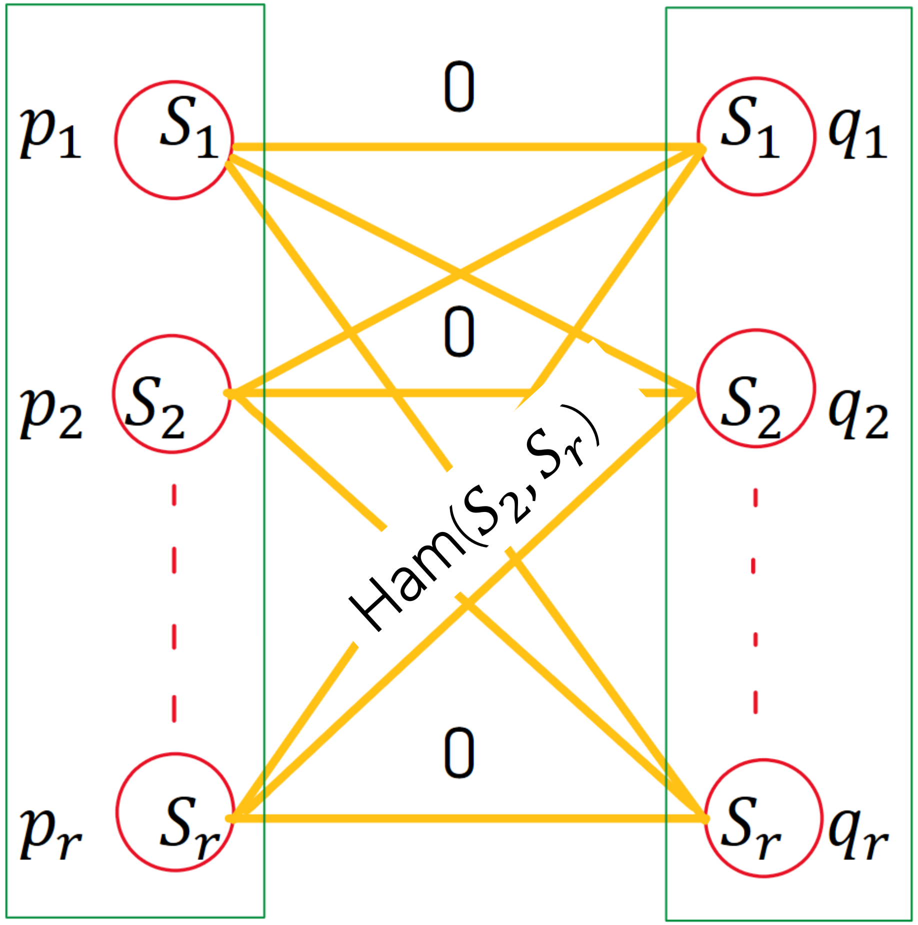
Distribution D_1 over solutions

Distribution D_2 over solutions

Average Sensitivity: Randomized Algorithms

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$$\mathrm{avg}_{e \in E}\left[\mathrm{d}_{\mathrm{EM}}\big(A(G), A(G-e)\big)\right]$$



Distribution D_1 over solutions

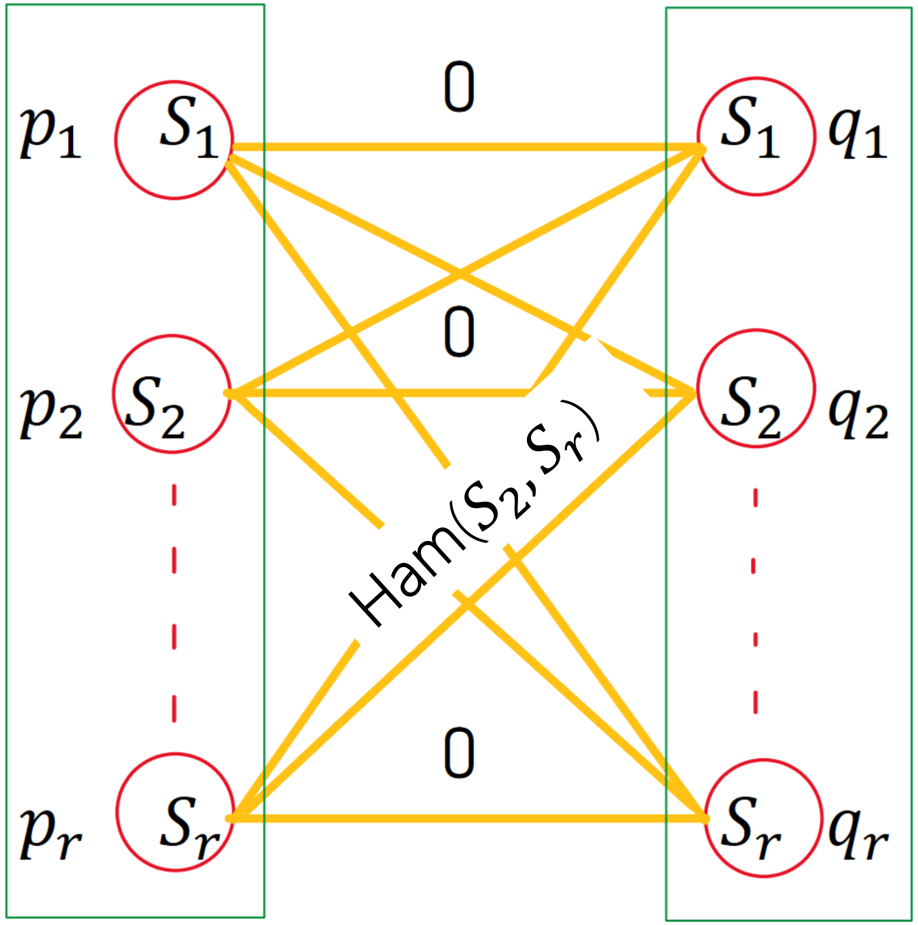
Distribution D_2 over solutions

Cost of moving prob. p from S_i to S_j is $p \cdot \mathrm{Ham}(S_i, S_j)$

Average Sensitivity: Randomized Algorithms

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$\mathrm{avg}_{e \in E} \left[\mathrm{d_{EM}} \left(A(G), A(G-e) \right) \right]$



Distribution D_1 over solutions

Distribution D_2 over solutions

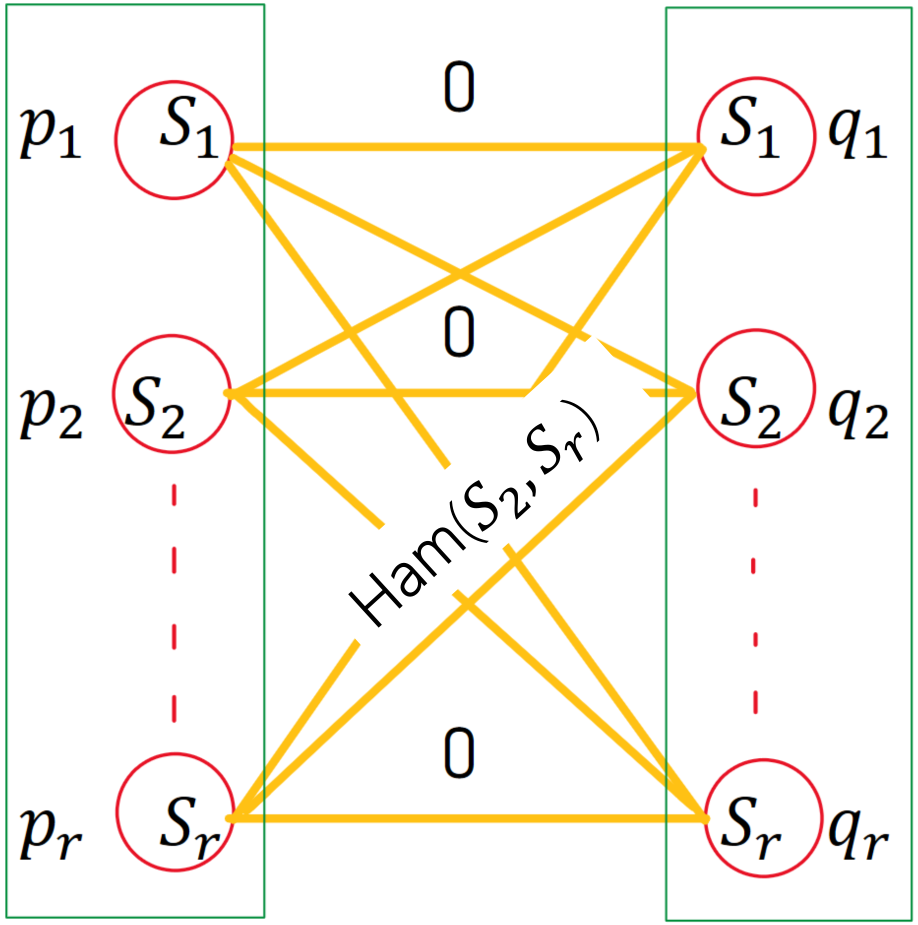
Optimal cost of moving the probability mass from one distribution to the other

Average Sensitivity: Randomized Algorithms

Average sensitivity of randomized algorithm A on graph $G = (V, E)$

$\text{avg}_{e \in E} \left[\text{d}_{\text{EM}}\big(A(G), A(G-e)\big) \right]$

Generalization to k-average sensitivity for the removal of k random edges (without replacement)



Distribution D_1 over solutions

Distribution D_2 over solutions

Optimal cost of moving the probability mass from one distribution to the other

Related Sensitivity Notions

- Differential Privacy [Dwork McSherry Nissim Smith '06]

Related Sensitivity Notions

- Differential Privacy [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm A on a graph G is differentially private if for all $e \in E$ the distributions $A(G)$ and $A(G - e)$ are close to each other

Related Sensitivity Notions

- **Differential Privacy** [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm A on a graph G is differentially private if for all $e \in E$ the distributions $A(G)$ and $A(G - e)$ are close to each other
  - Much stricter notion than average sensitivity

Related Sensitivity Notions

- Differential Privacy [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm $\boldsymbol{A}$ on a graph $\boldsymbol{G}$ is differentially private if for all $\boldsymbol{e} \in \boldsymbol{E}$ the distributions $\boldsymbol{A}(\boldsymbol{G})$ and $\boldsymbol{A}(\boldsymbol{G}-\boldsymbol{e})$ are close to each other
  - Much stricter notion than average sensitivity
  - Some of our algorithms inspired by differentially private algorithms

Related Sensitivity Notions

- Differential Privacy [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm A on a graph G is differentially private if for all $e \in E$ the distributions $A(G)$ and $A(G - e)$ are close to each other
  - Much stricter notion than average sensitivity
  - Some of our algorithms inspired by differentially private algorithms
- Stability of Learning Algorithms [Bousquet Elisseeff '02]

Related Sensitivity Notions

- **Differential Privacy** [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm $\boldsymbol{A}$ on a graph $\boldsymbol{G}$ is differentially private if for all $\boldsymbol{e} \in \boldsymbol{E}$ the distributions $\boldsymbol{A}(\boldsymbol{G})$ and $\boldsymbol{A}(\boldsymbol{G}-\boldsymbol{e})$ are close to each other
  - Much stricter notion than average sensitivity
  - Some of our algorithms inspired by differentially private algorithms
- **Stability of Learning Algorithms** [Bousquet Elisseeff '02]
  - A learner is stable if empirical loss does not change much by replacing any sample in the training data

Related Sensitivity Notions

- Differential Privacy [Dwork McSherry Nissim Smith '06]
  - Edge Differential Privacy [Nissim Raskhodnikova Smith '07]
    - An algorithm $\boldsymbol{A}$ on a graph $\boldsymbol{G}$ is differentially private if for all $\boldsymbol{e} \in \boldsymbol{E}$ the distributions $\boldsymbol{A}(\boldsymbol{G})$ and $\boldsymbol{A}(\boldsymbol{G}-\boldsymbol{e})$ are close to each other
  - Much stricter notion than average sensitivity
  - Some of our algorithms inspired by differentially private algorithms
- Stability of Learning Algorithms [Bousquet Elisseeff '02]
  - A learner is stable if empirical loss does not change much by replacing any sample in the training data
  - Stable learners have low generalization error

Talk Outline

- Our definition of average sensitivity for graph algorithms
- Key properties of our definition
- Main results
- Algorithm with low sensitivity for the global minimum cut problem
- Conclusions and open directions

k-Average Sensitivity from Average Sensitivity

Theorem: If A has average sensitivity $f(n, m)$, it has k-average sensitivity at most $\sum_{i \in [k]} f(n, m - i + 1)$.

Average Sensitivity Composes

Algorithms $\boldsymbol{A, B, C}$ such that $\boldsymbol{A(G) = B(G, C(G))}$

Average Sensitivity Composes

Algorithms A, B, C such that $A(G) = B(G, C(G))$

<u>Theorem</u> (Informal): Average sensitivity of A on $G = (V, E)$ can be bounded by the sum of:

- a term for average sensitivity of B, and
- a term for average sensitivity of C.

Average Sensitivity Composes

Algorithms A, B, C such that $A(G) = B(G, C(G))$

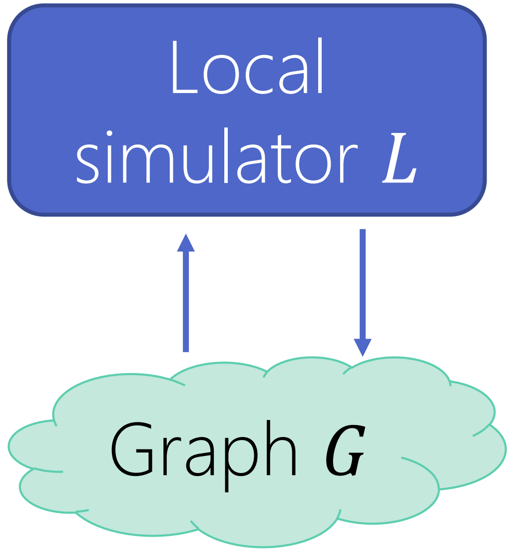
<u>Theorem</u> (Informal): Average sensitivity of A on $G = (V, E)$ can be bounded by the sum of:

- a term for average sensitivity of B, and
- a term for average sensitivity of C.

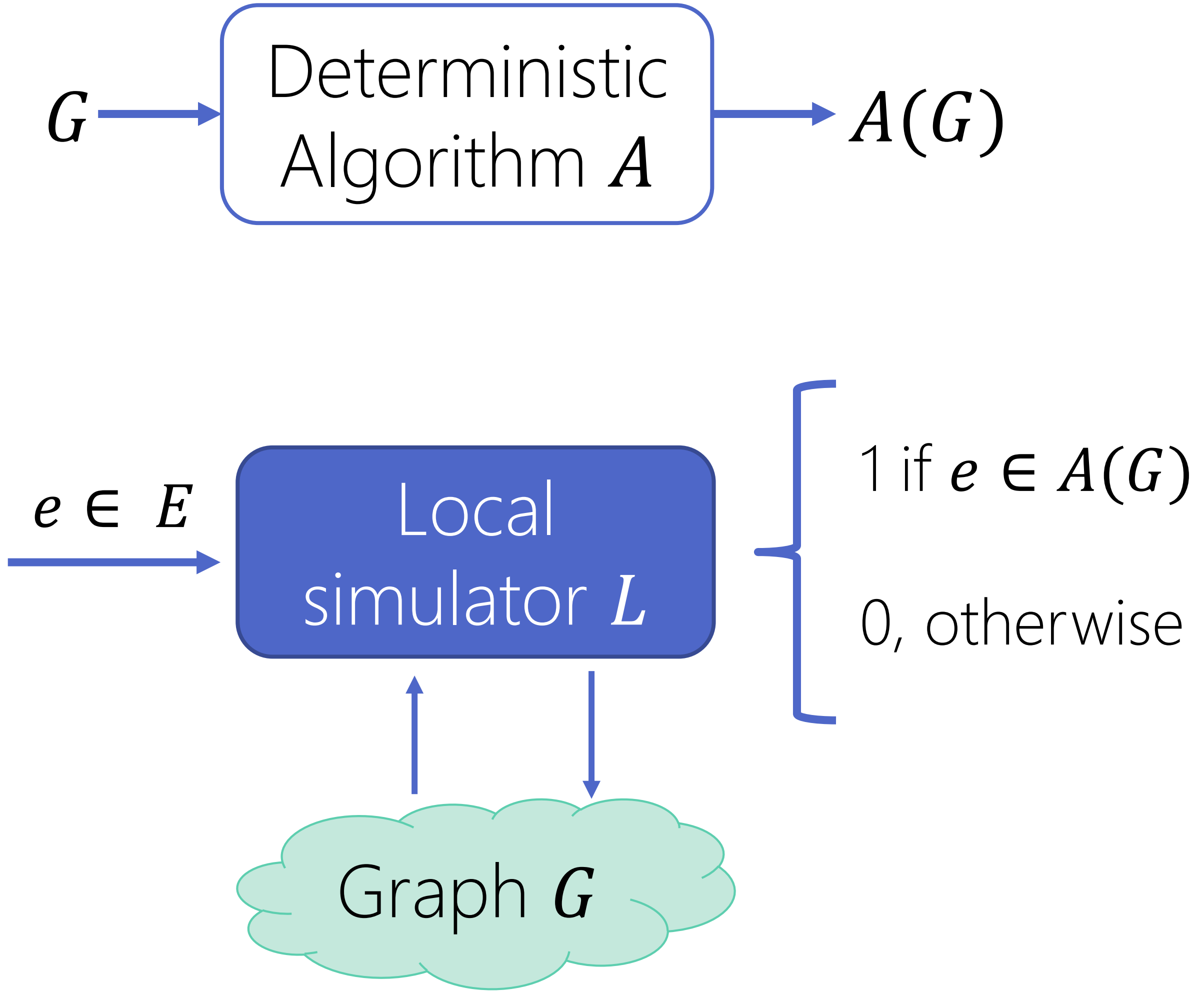
Can be used to bound the average sensitivity of a distribution over multiple stable-on-average algorithms.

Connection to Sublinear Algorithms

Connection to Sublinear Algorithms

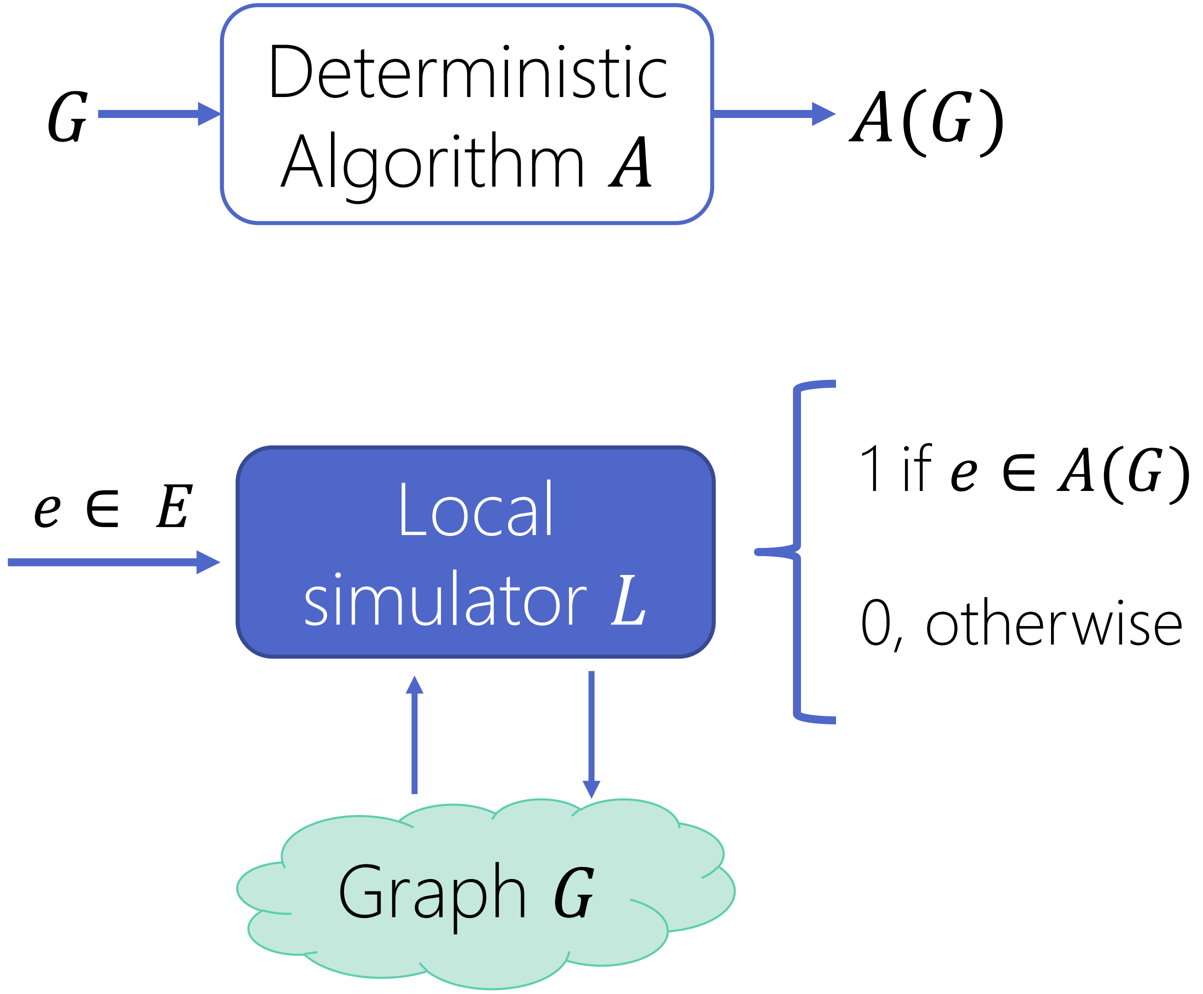


Connection to Sublinear Algorithms



Connection to Sublinear Algorithms

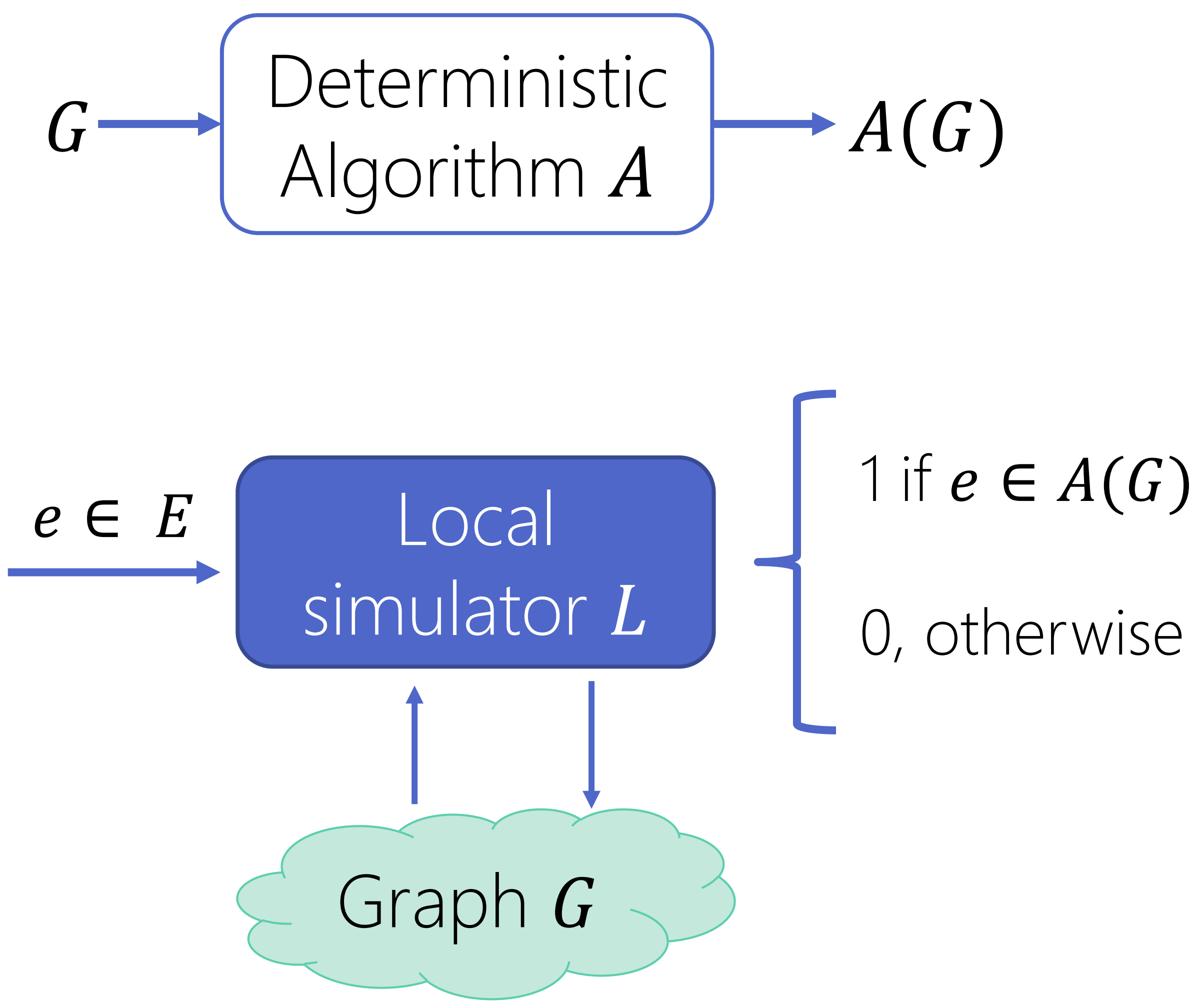
$$q(G) \triangleq \mathbb{E}_{e \in E}[\#\text{queries by } L]$$



Connection to Sublinear Algorithms

$q(G) \triangleq \mathbb{E}_{e \in E}[\#\text{queries by } L]$

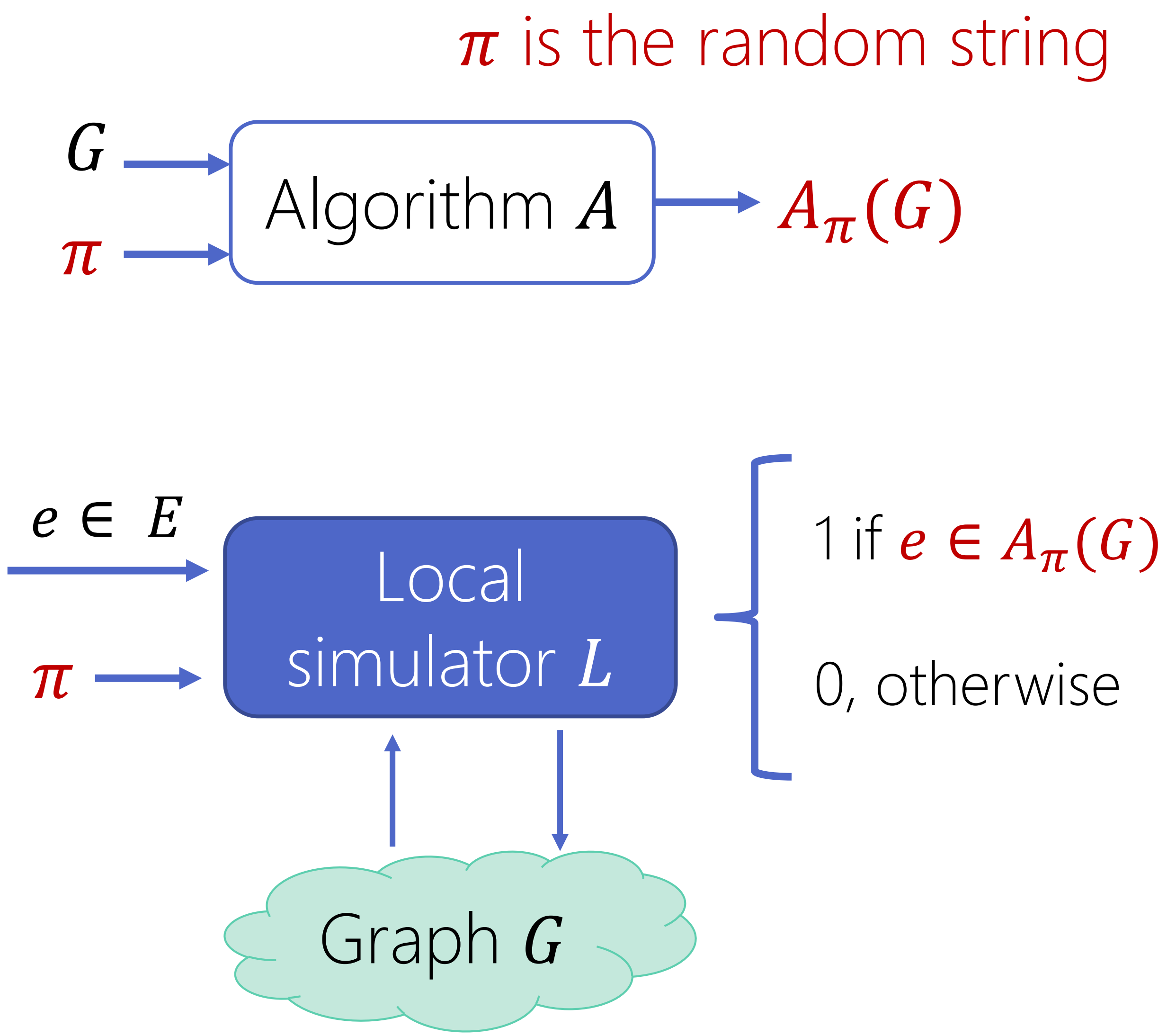
Average sensitivity of A on G is $\leq q(G)$



Connection to Sublinear Algorithms

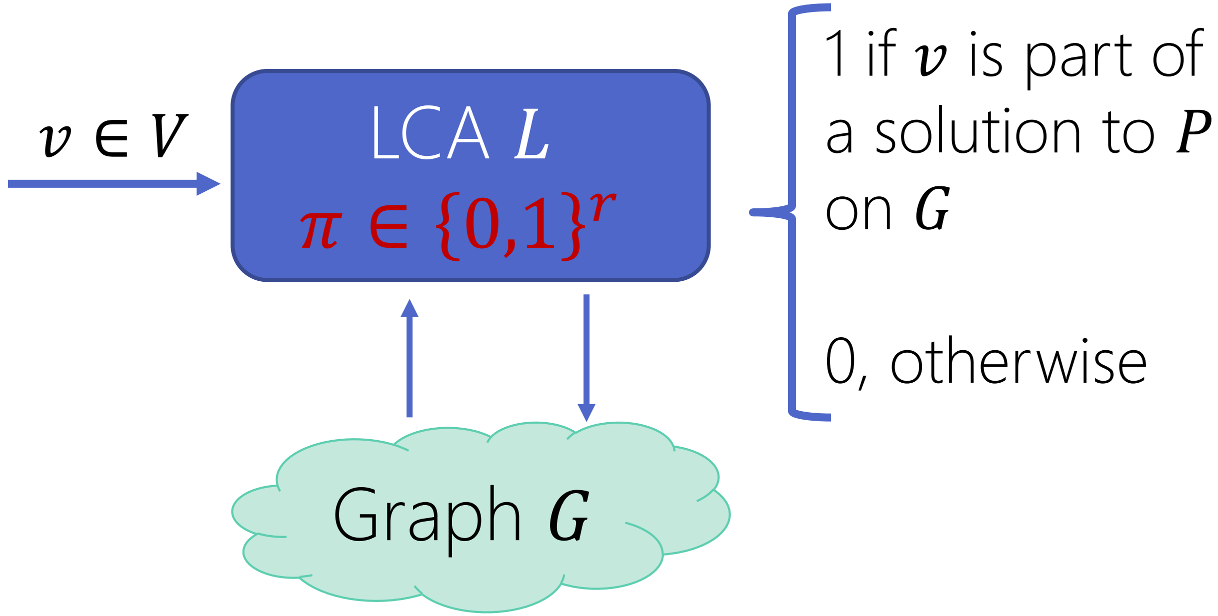
$q(G) \triangleq \mathbb{E}_{\pi, e \in E}[\#\text{queries by } L]$

Average sensitivity of A on G is $\leq q(G)$



Connection to Local Computation Algorithms (LCAs)

Graph problem P

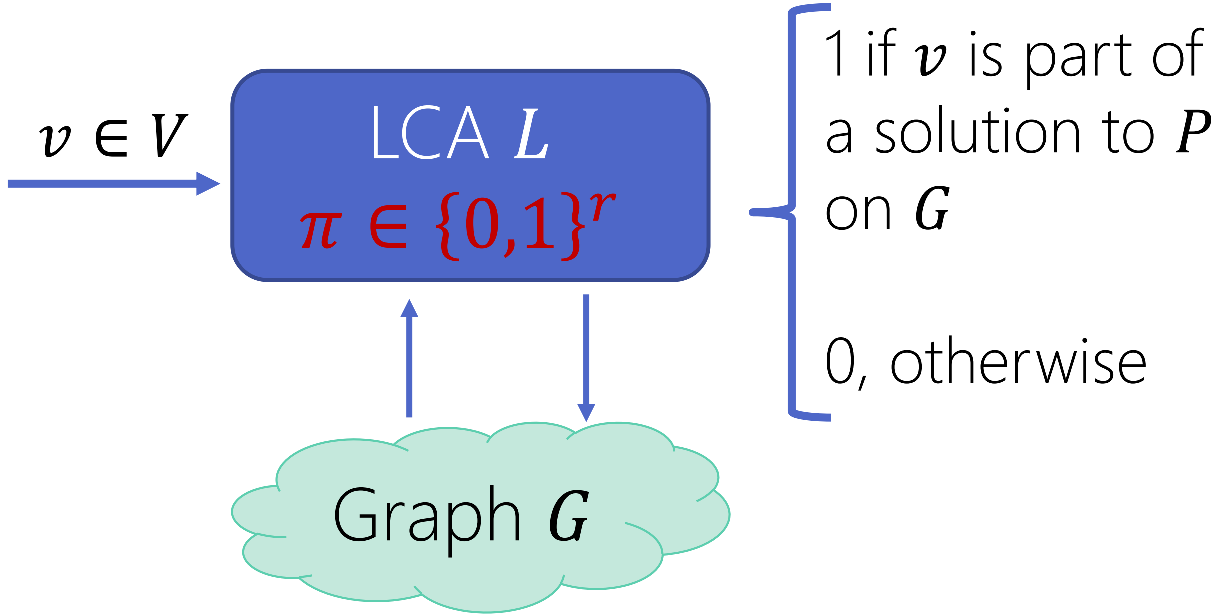


Answers of L are consistent with a single feasible solution of P on G

Connection to Local Computation Algorithms (LCAs)

If a problem P has an LCA of query complexity $q(G)$, then it has an algorithm with average sensitivity $\leq q(G)$

Graph problem $\boldsymbol{P}$



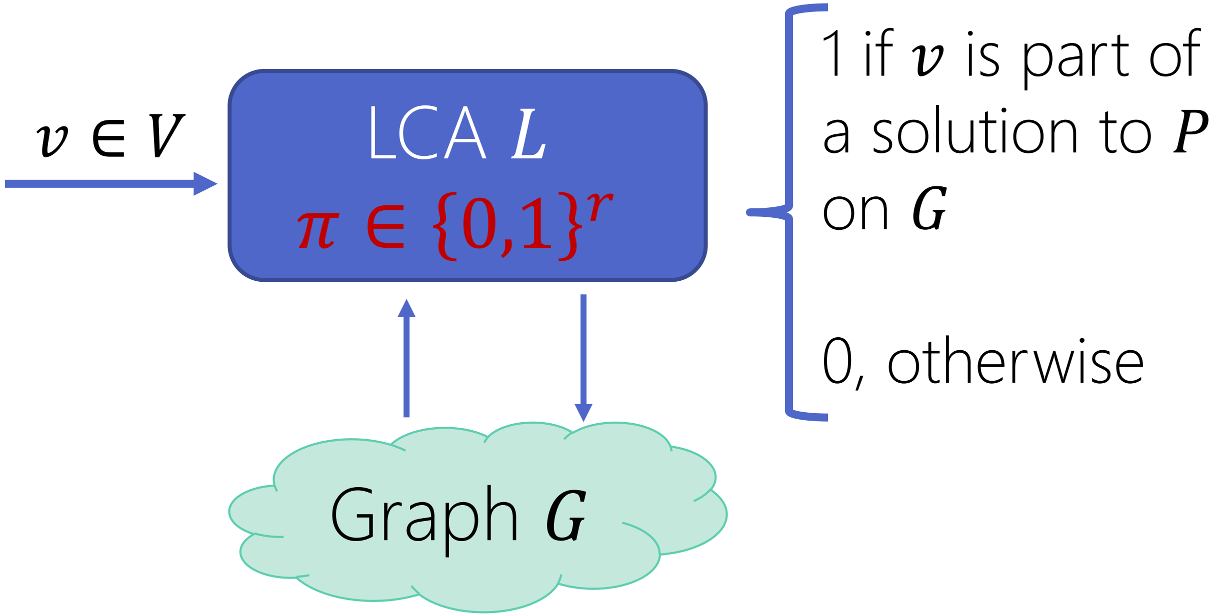
Answers of $\boldsymbol{L}$ are consistent with a single feasible solution of $\boldsymbol{P}$ on $\boldsymbol{G}$

Connection to Local Computation Algorithms (LCAs)

If a problem P has an LCA of query complexity $q(G)$, then it has an algorithm with average sensitivity $\leq q(G)$

Lower bound on average sensitivity implies lower bound on LCA query complexity!

Graph problem $\boldsymbol{P}$



Answers of $\boldsymbol{L}$ are consistent with a single feasible solution of $\boldsymbol{P}$ on $\boldsymbol{G}$

Talk Outline

- Our definition of average sensitivity for graph algorithms
- Key properties of our definition
- Main results
- Algorithm with low sensitivity for the global minimum cut problem
- Conclusions and open directions

Minimum Spanning Forest

For graphs on n vertices and m edges

Algorithm	Average Sensitivity
Kruskal's Algorithm	
Prim's Algorithm	

Minimum Spanning Forest

For graphs on n vertices and m edges

Algorithm	Average Sensitivity
Kruskal's Algorithm	$O(n/m)$
Prim's Algorithm	

Minimum Spanning Forest

For graphs on n vertices and m edges

Algorithm	Average Sensitivity
Kruskal's Algorithm	$O(n/m)$
Prim's Algorithm	$\Omega(n)$

For a specific tie-breaking rule

Other Problems We Study

- Maximum Cardinality Matching
  - Output an independent set of edges with maximum cardinality

Other Problems We Study

- **Maximum Cardinality Matching**
  - Output an independent set of edges with maximum cardinality
- **Global Minimum Cut**
  - Output a subset S of vertices with minimum number of edges between S and $V \setminus S$

Other Problems We Study

- Maximum Cardinality Matching
  - Output an independent set of edges with maximum cardinality
- Global Minimum Cut
  - Output a subset S of vertices with minimum number of edges between S and $V \setminus S$
- s-t Minimum Cut
- 2-Coloring

Maximum Cardinality Matching

For graphs on n vertices with max. matching size OPT

Maximum Cardinality Matching

For graphs on n vertices with max. matching size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$

Maximum Cardinality Matching

For graphs on n vertices with max. matching size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$1/2$	1

Maximum Cardinality Matching

For graphs on n vertices with max. matching size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$1/2$	1

Corollary: 2-approximation algorithm for minimum vertex cover with average sensitivity 2.

Maximum Cardinality Matching

For graphs on n vertices with max. matching size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$1/2$	1
$1-\epsilon$	$O\left(\left(\frac{OPT}{\epsilon^3}\right)^{\frac{1}{1+\epsilon^2}}\right)$

Corollary: 2-approximation algorithm for minimum vertex cover with average sensitivity $\mathbf{2}$.

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$2+\epsilon$	$n^{O(\frac{1}{\epsilon \mathrm{OPT}})}$

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$2+\epsilon$	$n^{O(\frac{1}{\epsilon \mathrm{OPT}})}$

If OPT = $\omega(\log n)$, average sensitivity is $O(1)$

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$2+\epsilon$	$n^{O(\frac{1}{\epsilon \mathrm{OPT}})}$
$< \infty$	$\Omega(n^{1/\mathrm{OPT}}/\mathrm{OPT}^2)$

If OPT = $\omega(\log n)$, average sensitivity is $O(1)$

Global Minimum Cut

For graphs on n vertices with global min. cut of size OPT

Approximation Ratio	Average Sensitivity
1	$\Omega(n)$
$2+\epsilon$	$n^{O(\frac{1}{\epsilon \mathrm{OPT}})}$
$< \infty$	$\Omega(n^{1/\mathrm{OPT}}/\mathrm{OPT}^2)$

If OPT = $\omega(\log n)$, average sensitivity is $O(1)$

If OPT = $O(\log n)$, average sensitivity is (nearly) optimal

s-t Minimum Cut

Problem: Given graph G and vertices s, t, find output a subset S of vertices with minimum number of edges between S and $V \setminus S$ such that $s \in S$ and $t \in V \setminus S$

Approximation (multiplicative, additive)	Average Sensitivity
$(1, O(n^{2/3}))$	$O(n^{2/3})$

2-Coloring

Problem: Given a bipartite graph G, output the set of vertices in one of the bipartitions.

Approximation (multiplicative, additive)	Average Sensitivity
–	$\Omega(n)$

Every LCA for $\mathbf{2}$-coloring has query complexity $\boldsymbol{\Omega(n)}$

Answers an open question raised by [Czumaj, Mansour, Vardi 18] on existence of sublinear-query LCAs for the problem of 2-coloring.

Talk Outline

- Our definition of average sensitivity for graph algorithms
- Key properties of our definition
- Main results
- Algorithm with low sensitivity for the global minimum cut problem
- Conclusions and Open directions

Global Minimum Cut Problem

Given $G = (V, E)$ and $S \subseteq V$,

size(S, G): number of edges crossing $(S, V \setminus S)$

Global Minimum Cut Problem

Given $G = (V, E)$ and $S \subseteq V$,

size(S, G): number of edges crossing $(S, V \setminus S)$

Problem: Output set $S \subseteq V$ with the minimum size.

Global Minimum Cut Problem

Given $G = (V, E)$ and $S \subseteq V$,

size(S, G): number of edges crossing $(S, V \setminus S)$

Problem: Output set $S \subseteq V$ with the minimum size.

Polynomial time exact algorithms exist.

Global Minimum Cut Problem

Given $G = (V, E)$ and $S \subseteq V$,

size(S, G): number of edges crossing $(S, V \setminus S)$

Problem: Output set $S \subseteq V$ with the minimum size.

Polynomial time exact algorithms exist.

Theorem [Karger 93]: For $\alpha \geq 1$, the number of cuts of size at most $\alpha \cdot \mathrm{OPT}$ is at most $n^{2\alpha}$ and they can be enumerated in polynomial time (per cut).

Global Minimum Cut

Theorem: There exists a polynomial time $(2+\epsilon)$-approximation algorithm with average sensitivity $n^{O\left(\frac{1}{\epsilon \mathrm{OPT}}\right)}$ for the global minimum cut problem for all $\epsilon > 0$.

Stable Algorithm for Global Minimum Cut

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;

Stable Algorithm for Global Minimum Cut

On input $\boldsymbol{G = (V, E)}$ and parameter $\boldsymbol{\epsilon > 0}$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;

Stable Algorithm for Global Minimum Cut

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Enumerate all cuts of size at most $(2 + \epsilon) \cdot \text{OPT}$;

Stable Algorithm for Global Minimum Cut

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Enumerate all cuts of size at most $(2 + \epsilon) \cdot \text{OPT}$;
- Output a cut $S \subseteq V$ with probability proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Stable Algorithm for Global Minimum Cut

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Enumerate all cuts of size at most $(2 + \epsilon) \cdot$ OPT;
- Output a cut $S \subseteq V$ with probability proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Sampling from an approximate Gibbs distribution

Stable Algorithm for Global Minimum Cut

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Enumerate all cuts of size at most $(2+\epsilon) \cdot \text{OPT}$;
- Output a cut $S \subseteq V$ with probability proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Sampling from an approximate Gibbs distribution

Inspired from a differentially private algorithm for global minimum cut [Gupta Ligett McSherry Roth Talwar '10]

Analysis

Approximation Ratio	Clear from algorithm description
Running time	Follows from Karger's theorem
Average Sensitivity	Will analyze now

Analysis: A (Slightly) Different Algorithm

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Sampling from Gibbs distribution

Analysis: A (Slightly) Different Algorithm

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Sampling from Gibbs distribution

Observation: Enough to bound average sensitivity of above *inefficient* algorithm, since its output distribution is close to original algorithm

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

Denote the inefficient algorithm using A

- Average sensitivity = Average (over $e \in E$) earth mover's distance between $A(G)$ and $A(G - e)$

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

$p(S, G)$:Probability that A outputs cut S on input G

Fix $e \in E$.

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

$p(S, G)$:Probability that A outputs cut S on input G

Fix $e \in E$.

- For cuts S such that e crosses S, $p(S, G - e) \approx p(S, G) \cdot \exp(\alpha)$

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

$p(S, G)$:Probability that A outputs cut S on input G

Fix $e \in E$.

- For cuts S such that e crosses S, $p(S, G - e) \approx p(S, G) \cdot \exp(\alpha)$
- Earth mover's distance between $A(G)$ and $A(G - e)$

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

$p(S, G)$:Probability that A outputs cut S on input G

Fix $e \in E$.

- For cuts S such that e crosses S, $p(S, G - e) \approx p(S, G) \cdot \exp(\alpha)$
- Earth mover's distance between $A(G)$ and $A(G - e)$

$$\approx n \cdot \sum_{S:e \text{ crosses } S} p(S, G - e) - p(S, G)$$

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

$p(S, G)$:Probability that A outputs cut S on input G

Fix $e \in E$.

- For cuts S such that e crosses S, $p(S, G - e) \approx p(S, G) \cdot \exp(\alpha)$
- Earth mover's distance between $A(G)$ and $A(G - e)$

$$\approx n \cdot \sum_{S: e \text{ crosses } S} p(S, G - e) - p(S, G)$$

$$= n \cdot (\exp \alpha - 1) \cdot \sum_{S: e \text{ crosses } S} p(S, G)$$

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \mathrm{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \mathrm{size}(S, G))$

- Average sensitivity of A is

$$\approx \frac{n}{m} \cdot (\exp \alpha - 1) \cdot \sum_{e} \sum_{S: e \text{ crosses } S} p(S, G)$$

(More Detailed Analysis Overview)

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \mathrm{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \mathrm{size}(S, G))$

- Average sensitivity of A is
$$\approx \frac{n}{m} \cdot (\exp \alpha - 1) \cdot \sum_{e} \sum_{S:e \text{ crosses } S} p(S, G)$$
- Average sensitivity of A is $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by A)

(More Detailed Analysis Overview)

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

- Average sensitivity of A is $$\approx \frac{n}{m} \cdot (\exp \alpha - 1) \cdot \sum_{e} \sum_{S: e \text{ crosses } S} p(S, G)$$
- Average sensitivity of A is $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by A)
- Expected size of cut $\leq (2 + \epsilon) \cdot \text{OPT} + o(1)$

(More Detailed Analysis Overview)

Analysis Overview

On input $G = (V, E)$ and parameter $\epsilon > 0$:

- Compute the value OPT;
- Let $\alpha \leftarrow \theta(\frac{\log n}{\epsilon \text{OPT}})$;
- Output cut $S \subseteq V$ with prob. proportional to $\exp(-\alpha \cdot \text{size}(S, G))$

- Average sensitivity of A is
$$\approx \frac{n}{m} \cdot (\exp \alpha - 1) \cdot \sum_{e} \sum_{S:e \text{ crosses } S} p(S, G)$$
- Average sensitivity of A is $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by A)
- Expected size of cut $\leq (2 + \epsilon) \cdot \text{OPT} + o(1)$
- $\text{OPT} \leq \frac{2m}{n}$, as min. cut size at most average degree

(More Detailed Analysis Overview)

Global Minimum Cut

Theorem: There exists a polynomial time $(2+\epsilon)$-approximation algorithm with average sensitivity $n^{O\left(\frac{1}{\epsilon \mathrm{OPT}}\right)}$ for the global minimum cut problem for all $\epsilon > 0$.

Global Minimum Cut

Theorem: There exists a polynomial time $(2+\epsilon)$-approximation algorithm with average sensitivity $n^{O\left(\frac{1}{\epsilon \mathrm{OPT}}\right)}$ for the global minimum cut problem for all $\epsilon > 0$.

Sampling from
Gibbs distribution
gives stability

Talk Outline

- Our definition of average sensitivity for graph algorithms
- Key properties of our definition
- Main results
- Algorithm with low sensitivity for the global minimum cut problem
- Conclusions and open directions

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties
- Design of stable algorithms for various combinatorial problems

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties
- Design of stable algorithms for various combinatorial problems
- Techniques for design of stable algorithms:

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties
- Design of stable algorithms for various combinatorial problems
- Techniques for design of stable algorithms:
  - Sampling from Gibbs distribution (Global Mincut)

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties
- Design of stable algorithms for various combinatorial problems
- Techniques for design of stable algorithms:
  - Sampling from Gibbs distribution (Global Mincut)
  - Notion of average sensitivity for LPs and stable LP solvers (s-t Mincut)

Summary of our contributions

- Introduced a definition of sensitivity of graph algorithms with several useful properties
- Design of stable algorithms for various combinatorial problems
- Techniques for design of stable algorithms:
  - Sampling from Gibbs distribution (Global Mincut)
  - Notion of average sensitivity for LPs and stable LP solvers (s-t Mincut)
  - Reusing analyses of existing sublinear-time algorithms and dynamic algorithms (Maximum Matching & Min. Vertex Cover)

Open Directions

- Stable-on-average algorithms for other combinatorial problems

Open Directions

- Stable-on-average algorithms for other combinatorial problems
  - Further applications of our techniques for design of stable algorithms

Open Directions

- Stable-on-average algorithms for other combinatorial problems
  - Further applications of our techniques for design of stable algorithms
- New stable-on-average procedures for building stable algorithms

Open Directions

- Stable-on-average algorithms for other combinatorial problems
  - Further applications of our techniques for design of stable algorithms
- New stable-on-average procedures for building stable algorithms
- Average sensitivity analyses of existing approximation algorithms

Open Directions

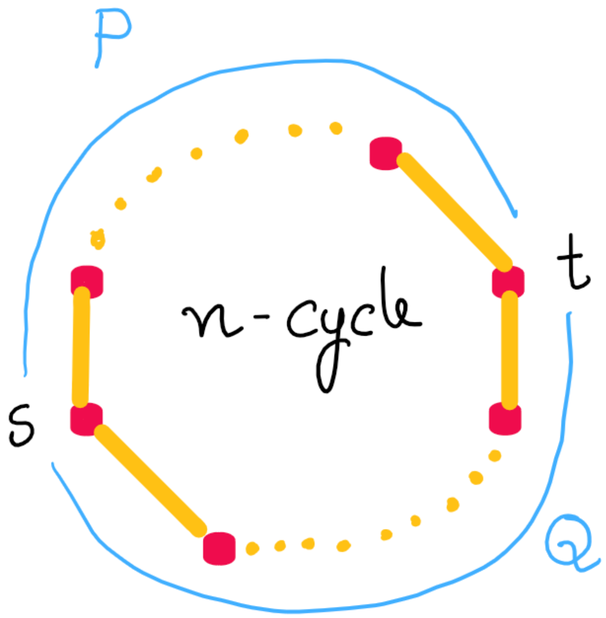
- Stable-on-average algorithms for other combinatorial problems
  - Further applications of our techniques for design of stable algorithms
- New stable-on-average procedures for building stable algorithms
- Average sensitivity analyses of existing approximation algorithms
- Average sensitivity lower bounds

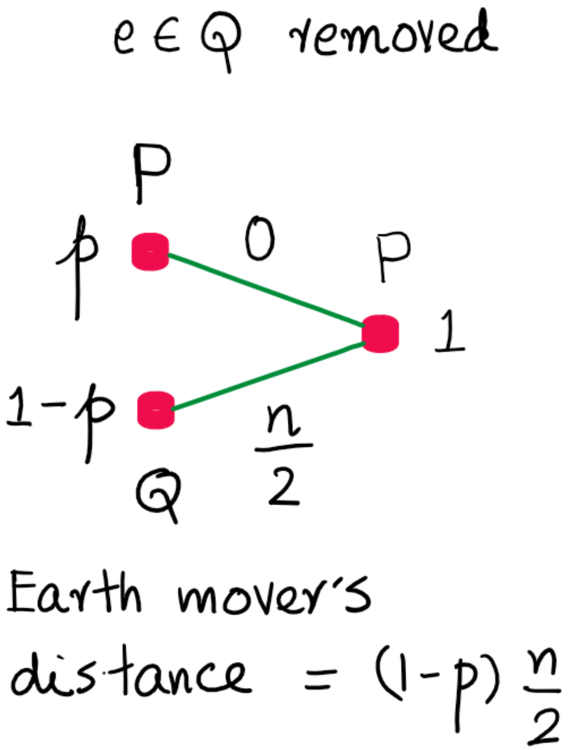
THANK YOU!

APPENDIX

Example 3: Average Sensitivity of s-t Shortest Path

Average sensitivity of outputting s-t shortest paths is $\Theta(n)$





P: output with probability p

Q: output with probability $1 - p$

Average sensitivity:

$$\frac{1}{2}\cdot(1-p)\cdot\frac{n}{2}+\frac{1}{2}\cdot p\cdot\frac{n}{2}$$

$$= \Omega(n)$$

Average Sensitivity Composes

- Algorithms A, B, C such that $A(G) = B(G, C(G))$
- H - Max. cardinality among solutions of A on n node graphs
- For $x \in C(G)$,
  $\text{Sens}_B(G, x)$ - avg. sensitivity of algo. $B(\cdot, x)$ on G

<u>Theorem</u>: Average sensitivity of A on $G = (V, E)$ is at most:

$$\mathbb{E}_{x \sim C(G)}[\text{Sens}_B(G, x)] + \text{H} \cdot \text{avg}_{e \in E} \left[\text{d}_{\text{TV}}\big(C(G), C(G - e)\big) \right]$$

Analysis: Expected Size of Cut Output

Denote the inefficient algorithm using $\boldsymbol{A}$

- Expected size of cut output by $\boldsymbol{A}$ is at most $(2+\epsilon) \cdot \mathrm{OPT} + o(1)$.
  - **Proof Idea**: Total probability mass assigned to cuts of size more than $(2+\epsilon) \cdot \mathrm{OPT}$ is $o(1)$.

Analysis: Average Sensitivity

- $Z = \sum_{T \subseteq V} \exp(-\alpha \cdot \text{size}(T, G))$;
- Z_e defined similarly;
- Probability that A outputs cut S on input G,

$$p(S, G) = \frac{\exp(-\alpha \cdot \text{size}(S, G))}{Z}$$

- For $e \in E$, $p(S, G) \cdot Z / Z_e \leq p(S, G - e)$

Claim: For $e \in E$, we have $\mathrm{d_{EM}}\big(A(G), A(G - e)\big) \leq n \cdot \left(\frac{Z_e}{Z} - 1\right)$

Analysis: Average Sensitivity

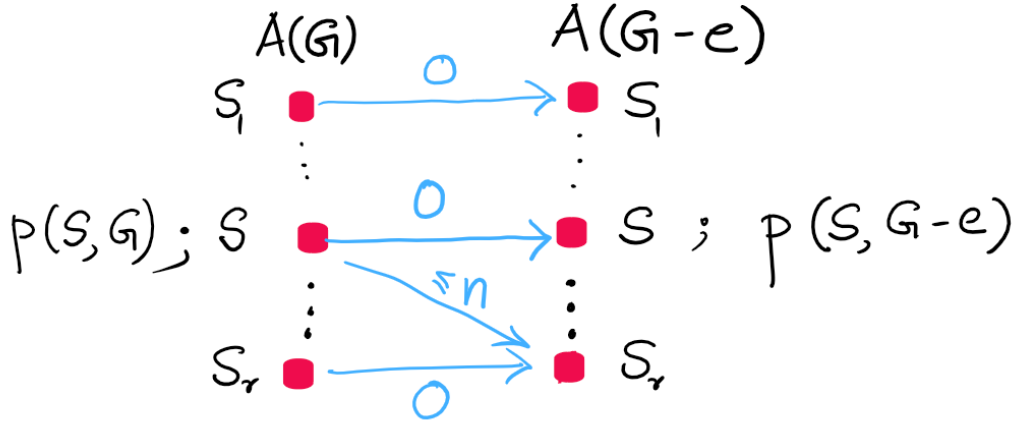
- $Z = \sum_{T \subseteq V} \exp(-\alpha \cdot \mathrm{size}(T, G))$
- $p(S, G) = \frac{\exp(-\alpha \cdot \mathrm{size}(S,G))}{Z}$
- $p(S, G) \cdot Z / Z_e \le p(S, G - e)$

Claim: For $e \in E$, we have

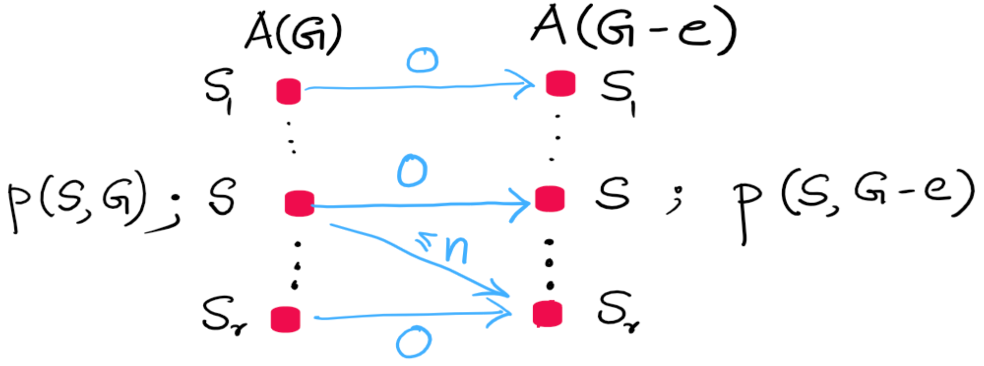
$$d_{\mathrm{EM}}(A(G), A(G-e)) \le n \cdot \left(\frac{Z_e}{Z} - 1\right)$$

Proof: Total Cost

$$\le n\left(1 - \frac{Z}{Z_e}\right) \le n\left(\frac{Z_e}{Z} - 1\right).$$



① Send $p(S,G) \cdot \frac{Z}{Z_e}$ at 0 cost



② Send $p(S,G) \cdot \left(1 - \frac{Z}{Z_e}\right)$ at

cost $\le n \cdot p(S,G) \cdot \left(1 - \frac{Z}{Z_e}\right)$

Total Cost $\le n\left(1 - \frac{Z}{Z_e}\right) \sum_S p(S,G) = n\left(1 - \frac{Z}{Z_e}\right)$

Analysis: Average Sensitivity

- **Claim:** Average sensitivity of $\boldsymbol{A}$ is
  $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by $\boldsymbol{A}$)
- **Proof:** Average sensitivity

$$\leq \frac{n}{m} \sum_{e \in E} \left(\frac{Z_e}{Z} - 1 \right) = \frac{n}{mZ} \sum_{e \in E} Z_e - Z$$

$$= \frac{n}{mZ} \sum_{e \in E} \sum_{\substack{S \subseteq V: \\ e \text{ crosses } S}} \left[\exp(-\alpha \cdot \text{size}(S, G-e)) - \exp(-\alpha \cdot \text{size}(S, G)) \right]$$

Analysis: Average Sensitivity

- **Claim:** Average sensitivity of $\boldsymbol{A}$ is
  $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by $\boldsymbol{A}$)
- Proof (contd.):

$$= \frac{n \cdot (\exp(\alpha) - 1)}{mZ} \cdot \sum_{e \in E} \sum_{\substack{S \subseteq V: \\ e \text{ crosses } S}} \exp(-\alpha \cdot \text{size}(S, G))$$

$$= \frac{n}{m} \cdot (\exp(\alpha) - 1) \cdot \sum_{S \subseteq V} \text{size}(S, G) \cdot \frac{\exp(-\alpha \cdot \text{size}(S, G))}{Z}$$

$$= \frac{n}{m} (\exp(\alpha) - 1) \cdot (\text{Expected size of cut output by } A)$$

Analysis: Average Sensitivity

- Average sensitivity of A is
  $\leq \frac{n}{m} \cdot (\exp \alpha - 1) \cdot$(Expected size of cut output by A)
- Expected size of cut output by $A \leq (2 + \epsilon) \cdot$ OPT + $o(1)$
- OPT $\leq \frac{2m}{n}$, as mincut size at most average degree
- $\alpha = \theta(\log n / \epsilon \text{OPT})$, by our setting

Theorem: Average sensitivity of A is $n^{O(1/\epsilon \text{OPT})}$.

Why not total variation distance?

- Consider algorithms A and B that output subsets of vertices
- Given a graph G, edge $e \in E$, $v \in V$ and $S \subseteq V$ be a set containing v
- $A(G) = S$ w.p. $\frac{3}{4}$ and $A(G) = S \setminus \{v\}$ w.p. $\frac{1}{4}$
  - $A(G - e) = S$ w.p. $\frac{1}{4}$ and $A(G - e) = S \setminus \{v\}$ w.p. $\frac{3}{4}$
  - TV distance ≤ 1
  - Earth mover's distance = 1

Why not total variation distance?

- Consider algorithms A and B that output subsets of vertices
- Given a graph G, edge $e \in E$, $v \in V$ and $S \subseteq V$ be a set containing v
- $B(G) = S$ w.p. $\frac{3}{4}$ and $B(G) = S \setminus \{v\}$ w.p. $\frac{1}{4}$
  - $B(G-e) = S$ w.p. $\frac{1}{4 \cdot 2^n}$, $B(G-e) = S \setminus \{v\}$ w.p. $\frac{3}{4} + \frac{1}{4 \cdot 2^n}$, and $B(G-e) = T$ w.p. $\frac{1}{4 \cdot 2^n}$
  - TV distance ≤ 1
  - Earth mover's distance = $\Omega(n)$