The Diameter of Randomly Perturbed Digraphs and 2 Applications

Abraham Flaxman
&
Alan Frieze
(CMU Math)

Introduction: Smoothed Analysis:

Introduction: Smoothed Analysis:

[Speilman + Teng, 2001]

Proposed in a theoretical explaination for the impressive performance of the simplex algorithm for linear programming.

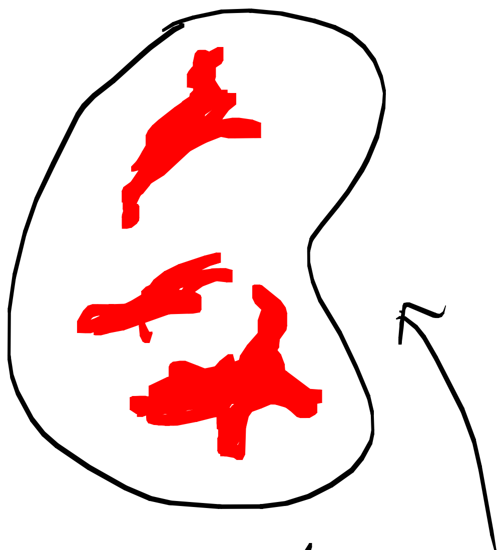
Introduction: Smoothed Analysis:

[Speilman + Teng, 2001]

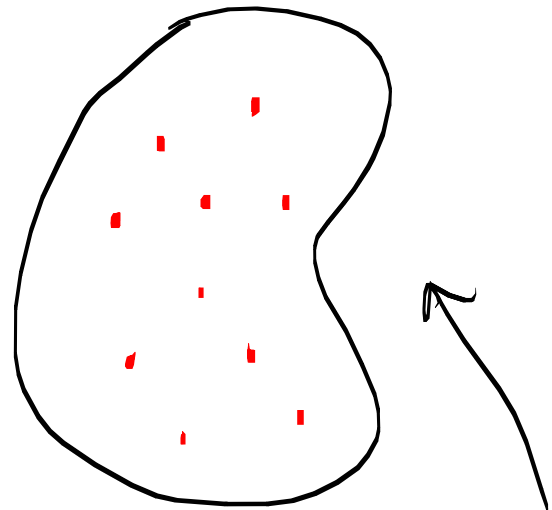
Proposed in a theoretical explaination for the impressive performance of the simplex algorithm for linear programming.

In short, define a random perturbation and show the hard instances are unlikely with respect to this perturbation.

Introduction: Smoothed Analysis:



Things like this should be difficult in practice



Things like this should be doable in practice

Introduction: Smoothed Analysis:

Is this a good theoretical explaination?

Introduction: Smoothed Analysis:

Is this a good theoretical explaination?

What does it predict for the perceptron algorithm for linear programming?

[Blum & Dunagan, 2002]

[Dunagan & Vempala, 2004]

Introduction: Smoothed Analysis:

Is this a good theoretical explaination?

What does it predict for the perceptron algorithm for linear programming?

[Blum & Dunagan, 2002]

[Dunagan & Vempala, 2004]

Is the perturbation "right"?

(In their case, additive noise, consisting of independent $\mathcal{N}(0, \varepsilon)$.)

Smoothed analysis of discrete problems

What is the appropriate perturbation if the input is a graph on n vertices?

Smoothed analysis of discrete problems

What is the appropriate perturbation if the input is a graph on n vertices?

Simple answer: XOR with a random graph.

Smoothed analysis of discrete problems

What is the appropriate perturbation if the input is a graph on n vertices?

Simple answer: XOR with a random graph.

but...

if random graph has too many edge, there will be nothing left of the original instance.

Smoothed analysis of discrete problems

(Acceptable in certain situations, for example, property testing [Speilman and Teng, 2002])

Smoothed analysis of discrete problems

(Acceptable in certain situations, for example, property testing [Speilman and Teng, 2002])

But we want to say something about sparse graphs.

So, we perturb by XORing with $G_{n,p}$ where $p = \frac{\varepsilon}{n}$.

Smoothed analysis of discrete problems

Relationship to semirandom instances.

Smoothed analysis of discrete problems

Relationship to semirandom instances.

Semirandomness

[Santha & Vazirani, 1986]

Semirandom instances

[Blum & Spencer, 1995]

Smoothed analysis of discrete problems

Relationship to semirandom instances.

Semirandomness

[Santha & Vazirani, 1986]

Semirandom instances

[Blum & Spencer, 1995]

To go from semirandom instances to smoothed instances, make advisary oblivious.

Central Observation:

Central Observation:

Theorem: Let $\varepsilon > 0$. Let $\overline{G}$ be an n-node connected graph. Let $G = \overline{G} + R$ where $R \sim G_{n, \varepsilon/n}$. Then, whp,

$$\operatorname{diam}(G) \le 100\,\varepsilon^{-1} \log n.$$

Central Observation:

Theorem: Let $\varepsilon > 0$. Let $\bar{G}$ be an n-node connected graph. Let $G = \bar{G} + R$ where $R \sim \mathbb{G}_{n, \varepsilon/n}$. Then, whp,

$$\mathrm{diam}(G) \le 100\, \varepsilon^{-1} \log n.$$

(In paper, stated for digraphs, but holds for directed/undirected, and for many similar perturbations)

Proof:

Proof: [Karp, 90]

We adapt a technique developed by Karp, and consider a algorithmic exploration of G that looks a lot like a branching process.

Proof: [Karp, 90]

We adapt a technique developed by Karp, and consider a algorithmic exploration of G that looks a lot like a branching process.

(Also similar to [Bollobás & Chung, 88] which shows diameter of a cycle + random matching is $O(\log n)$.)

Proof:

Fix s, t.

Proof:

Fix s, t.

Proof:

Fix s, t. We will show a path of length $O(\varepsilon^{-1} \log n)$ connects s and t whp.

Proof:

Fix s, t. We will show a path of length $O(\varepsilon^{-1} \log n)$ connects s and t whp.

To do so, we explore neighborhoods around s and t, alternating between edges of $\overline{G}$ and R.

Proof:

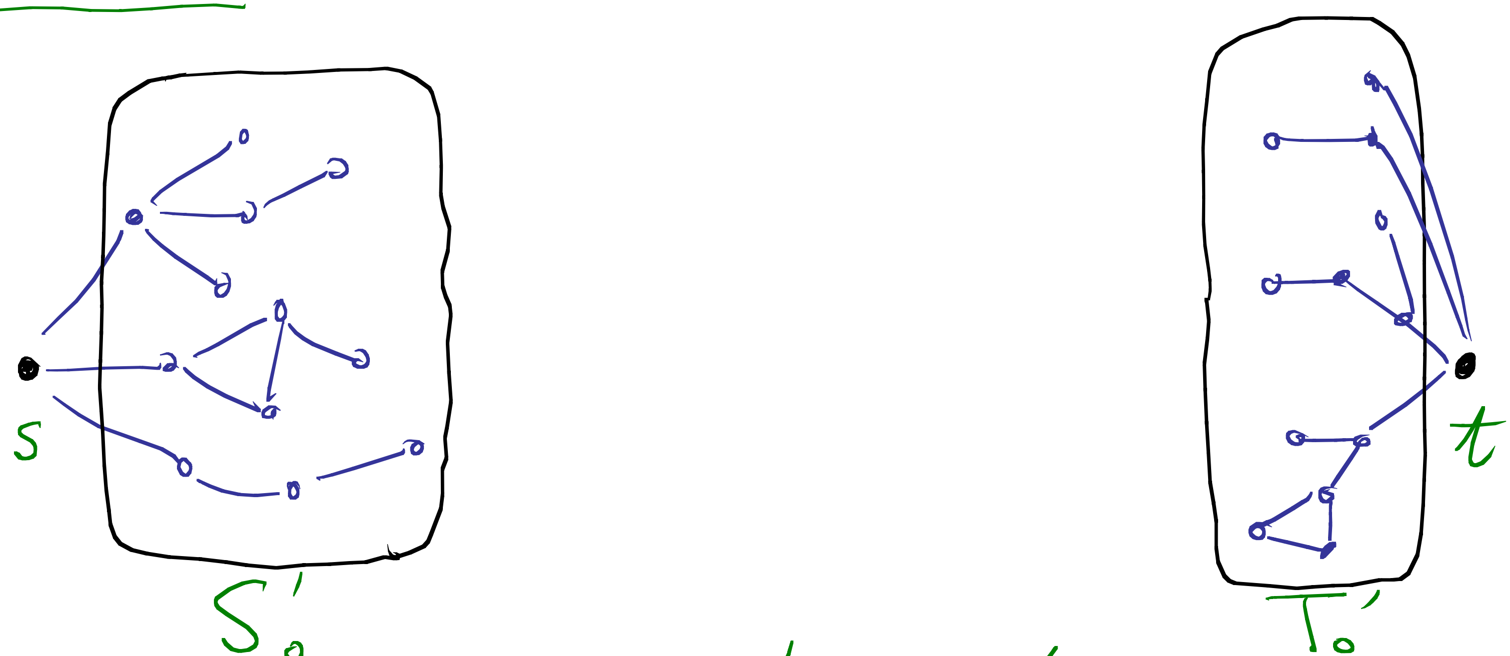
First, we find S_0' & T_0', where

$$S_0' = \{ v : d_{\bar{G}}(s,v) \leq O(\varepsilon^{-1} \log n) \}$$

and

T_0' is defined analogously.

Proof:



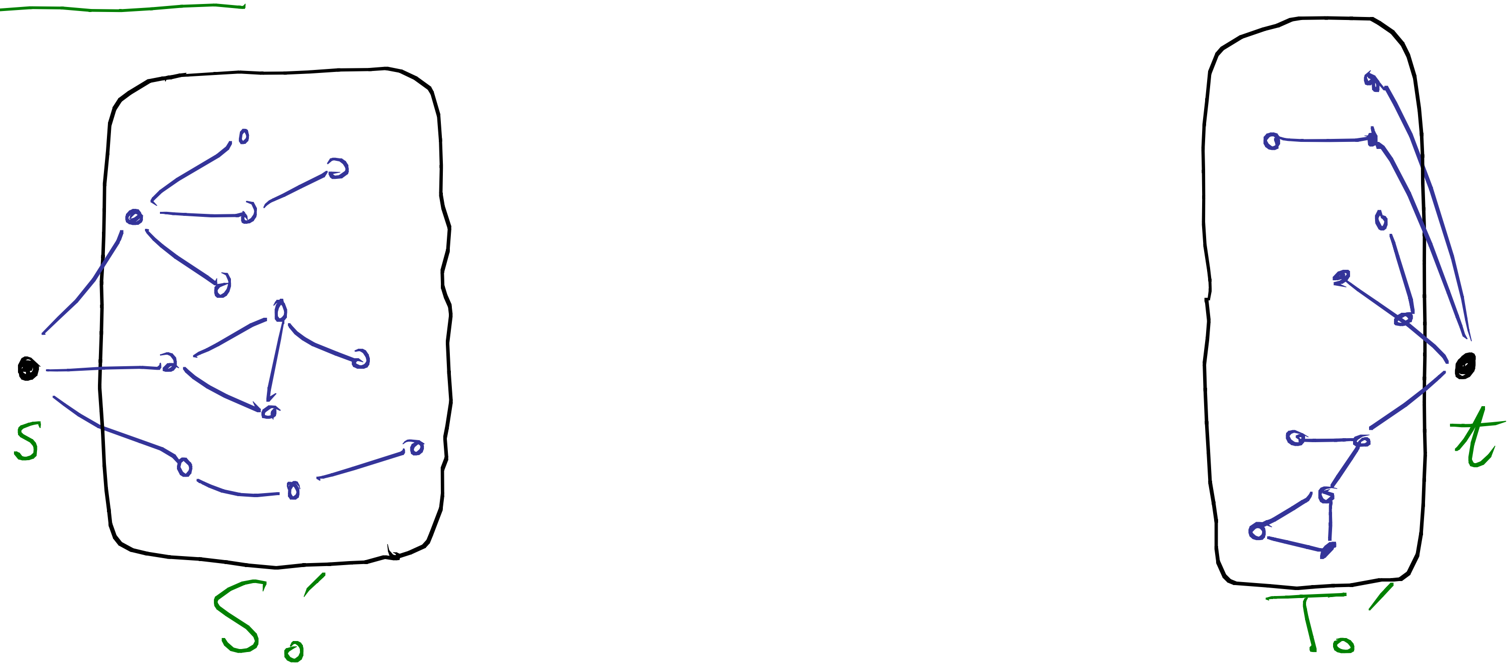
First, we find S_0' & T_0', where

$$S_0' = \{ v : d_{\bar{G}}(s,v) \le O(\varepsilon^{-1}\log n) \}$$

and

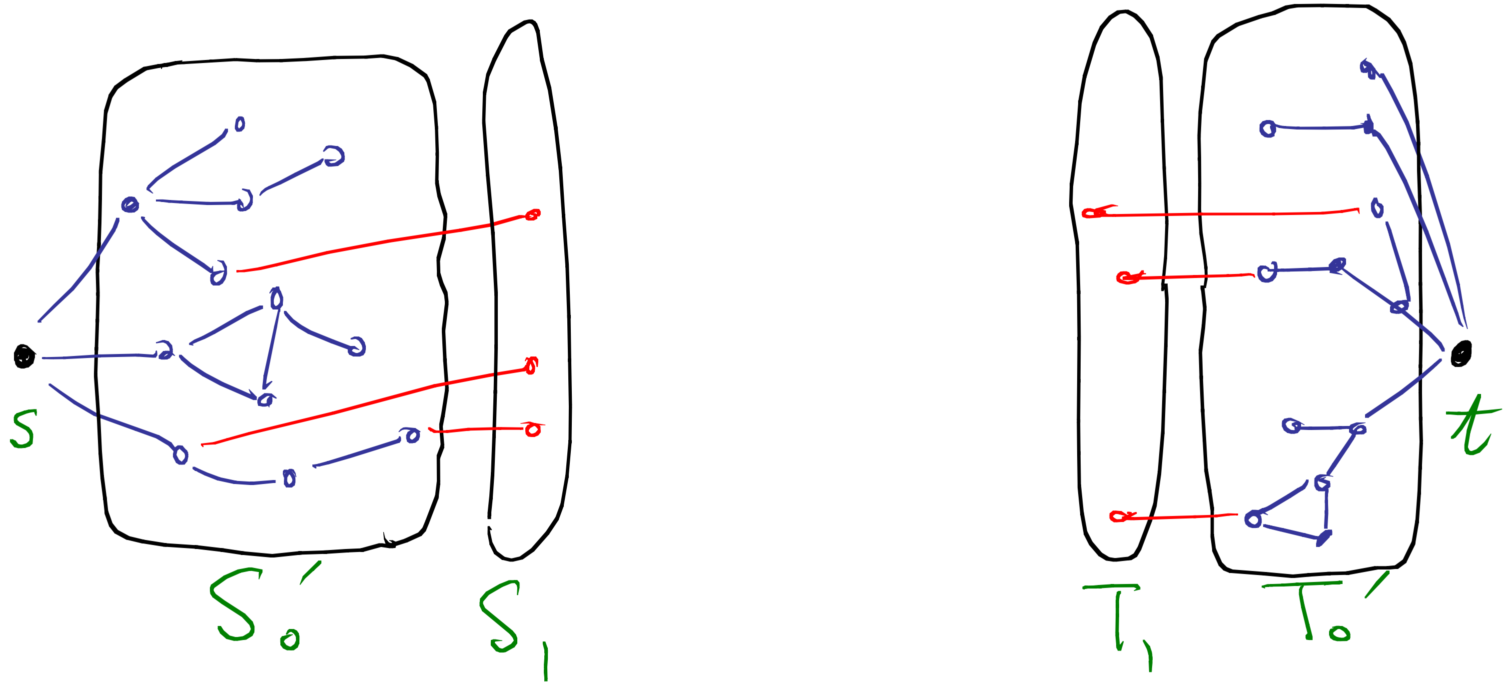
T_0' is defined analogously.

Proof:



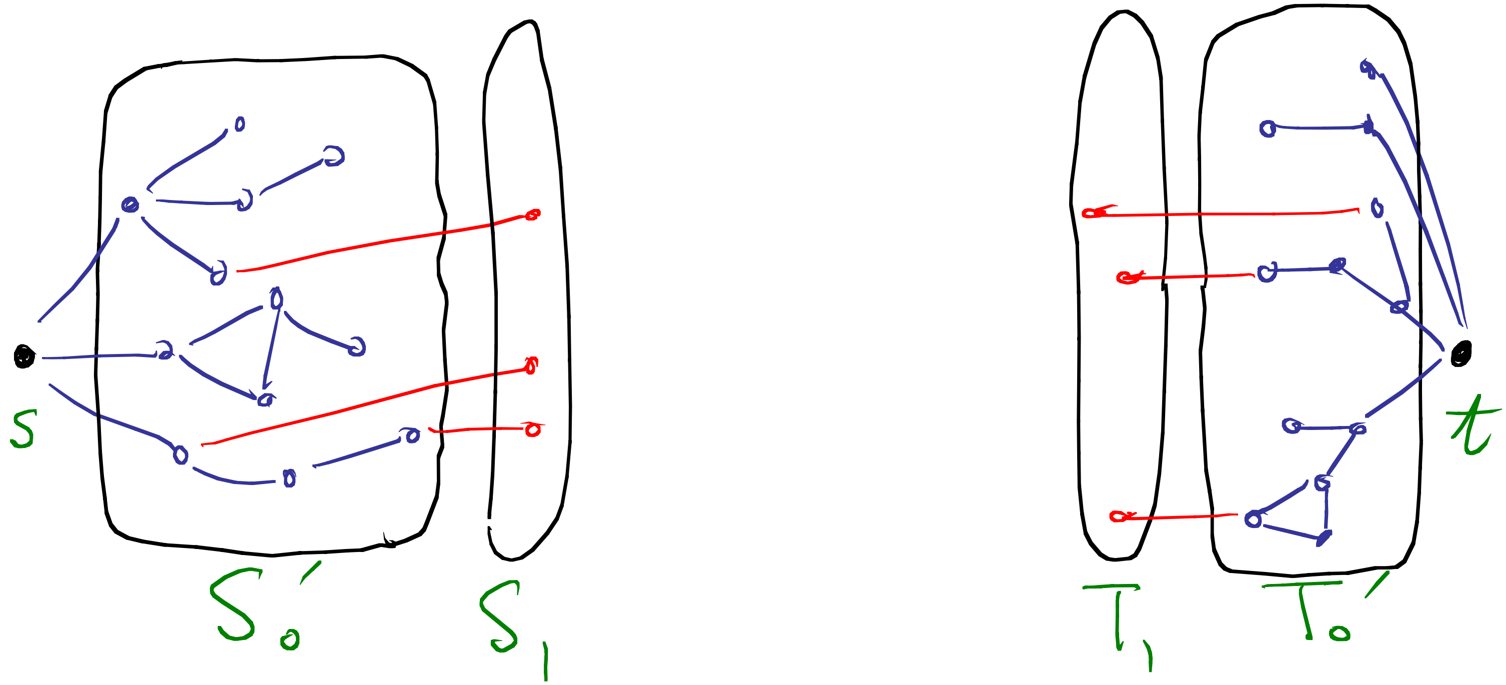
Then we reveal all the random edges of R incident to S_0' and T_0', and denote their other endpoints by S_1 & T_1.

Proof:



Then we reveal all the random edges of R incident to S_0' and T_0', and denote their other endpoints by S_1 & T_1.

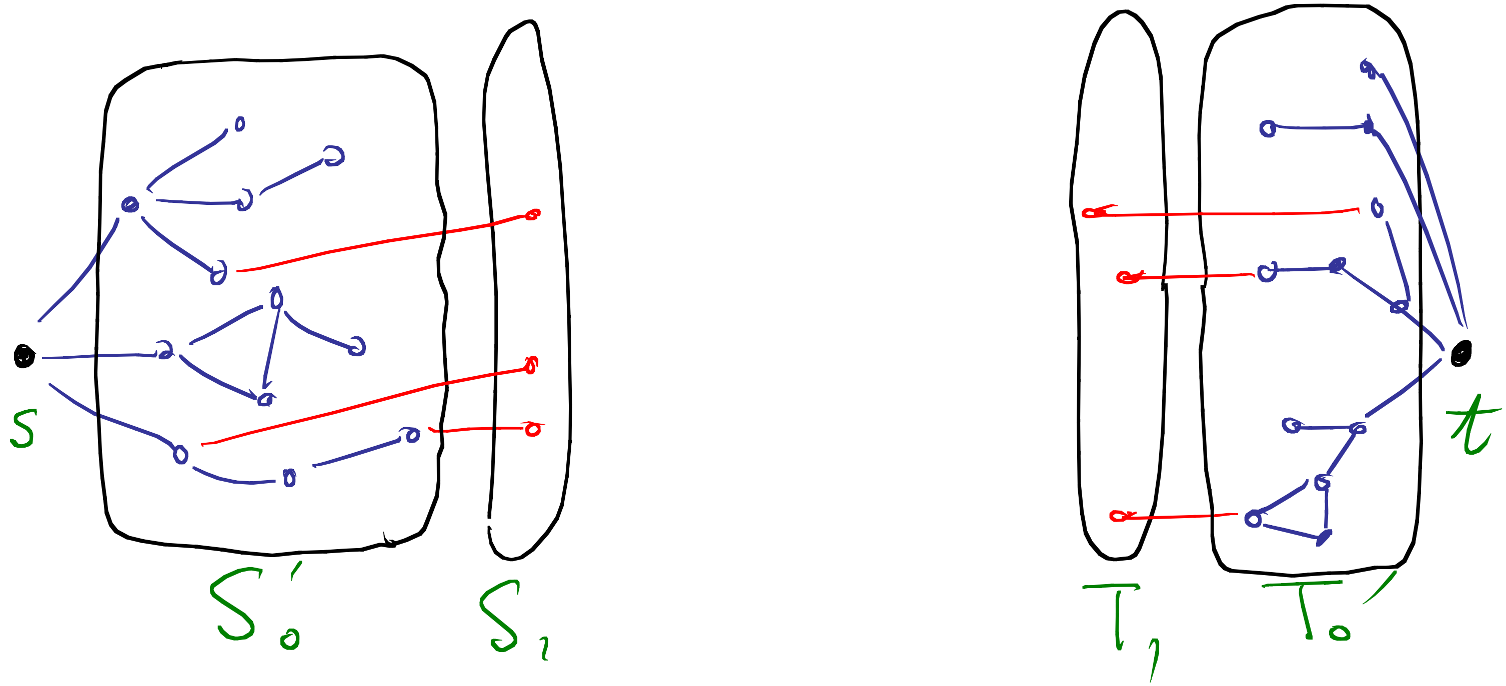
Proof:



Then we reveal all the random edges of R incident to S_0' and T_0', and denote their other endpoints by S_1 & T_1.

$$\mathbb{E}[|S_1|] \gtrsim |S_0'| \cdot n \cdot \frac{\varepsilon}{n} = \mathcal{O}(\log n).$$

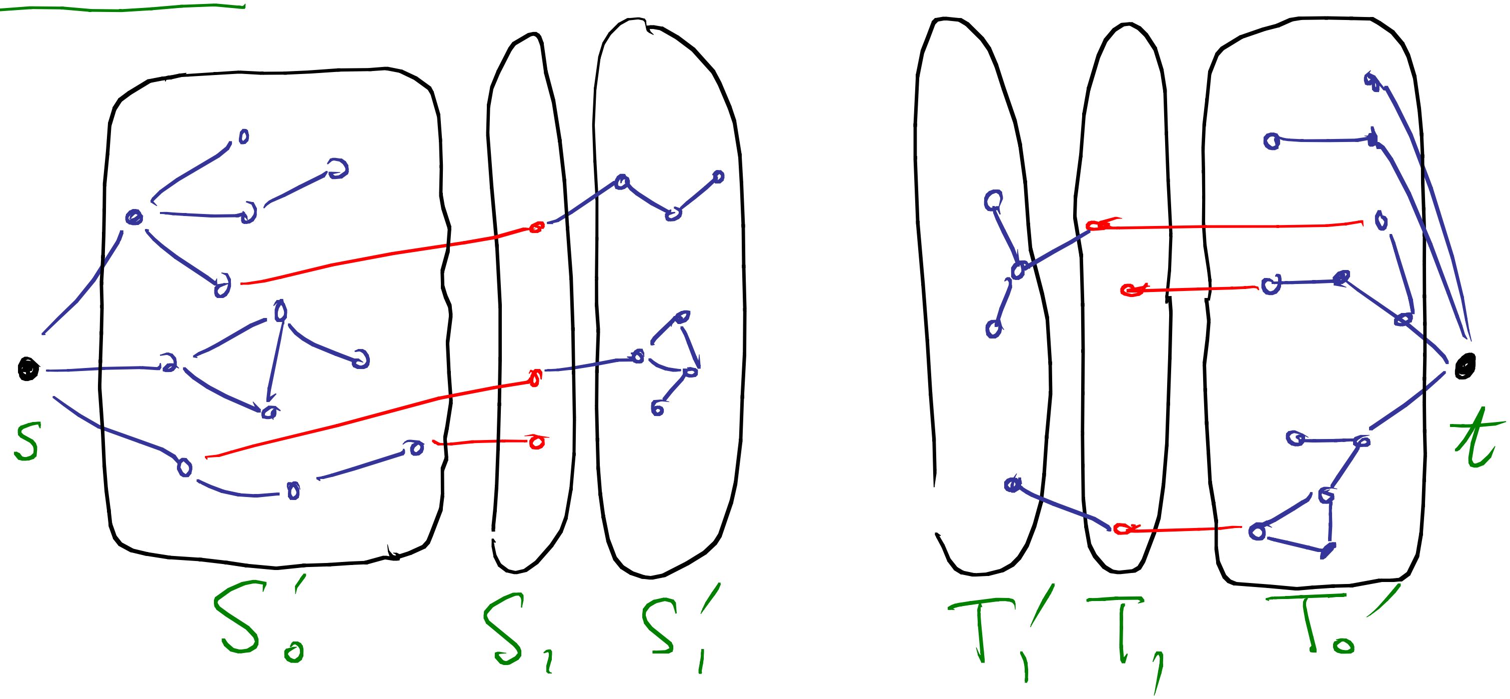
Proof:



Now, we let S_1', T_1' be defined by

$$S_1' = \{v : d_{\bar{G}}(S_0', v) \le 5\varepsilon^{-1}\}$$

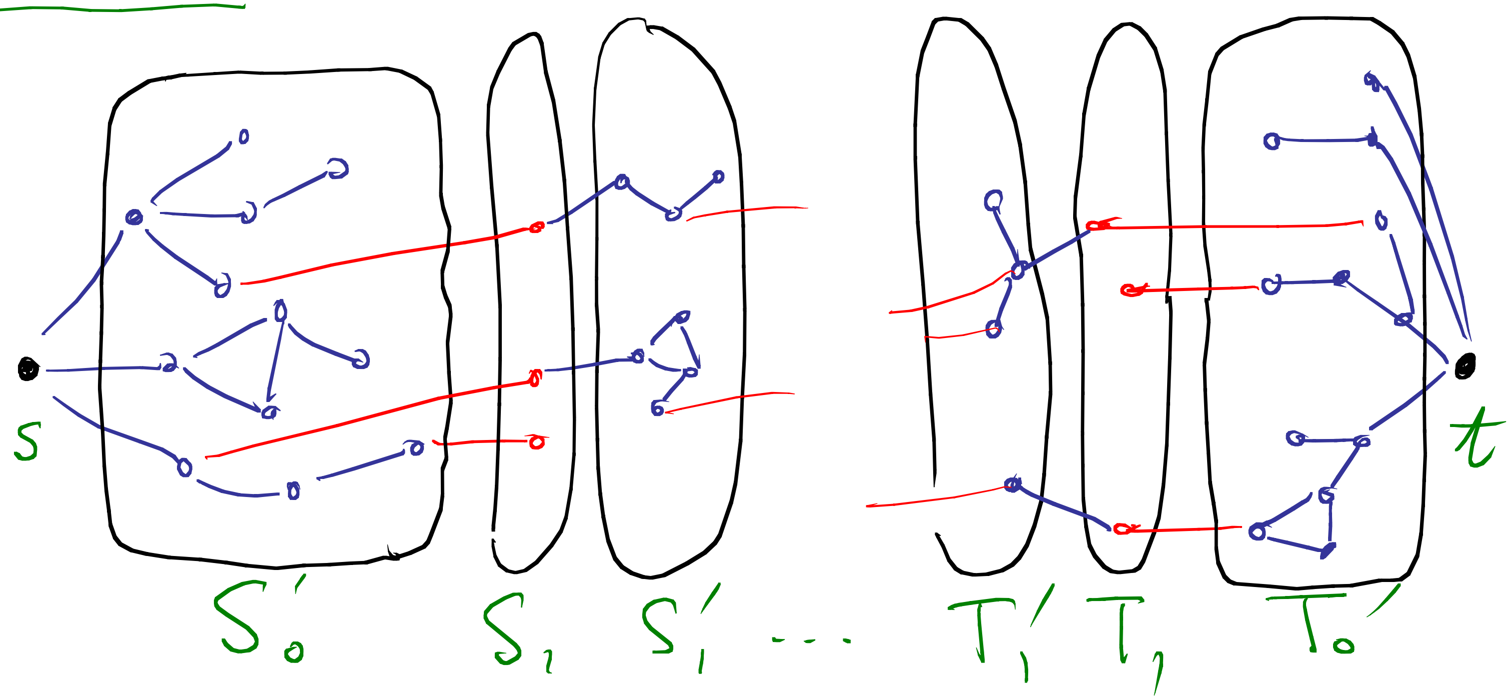
Proof:



Now, we let S_1', T_1' be defined by

$$S_1' = \{v : d_{\bar{G}}(S_0', v) \leq 5\varepsilon^{-1}\}$$

Proof:



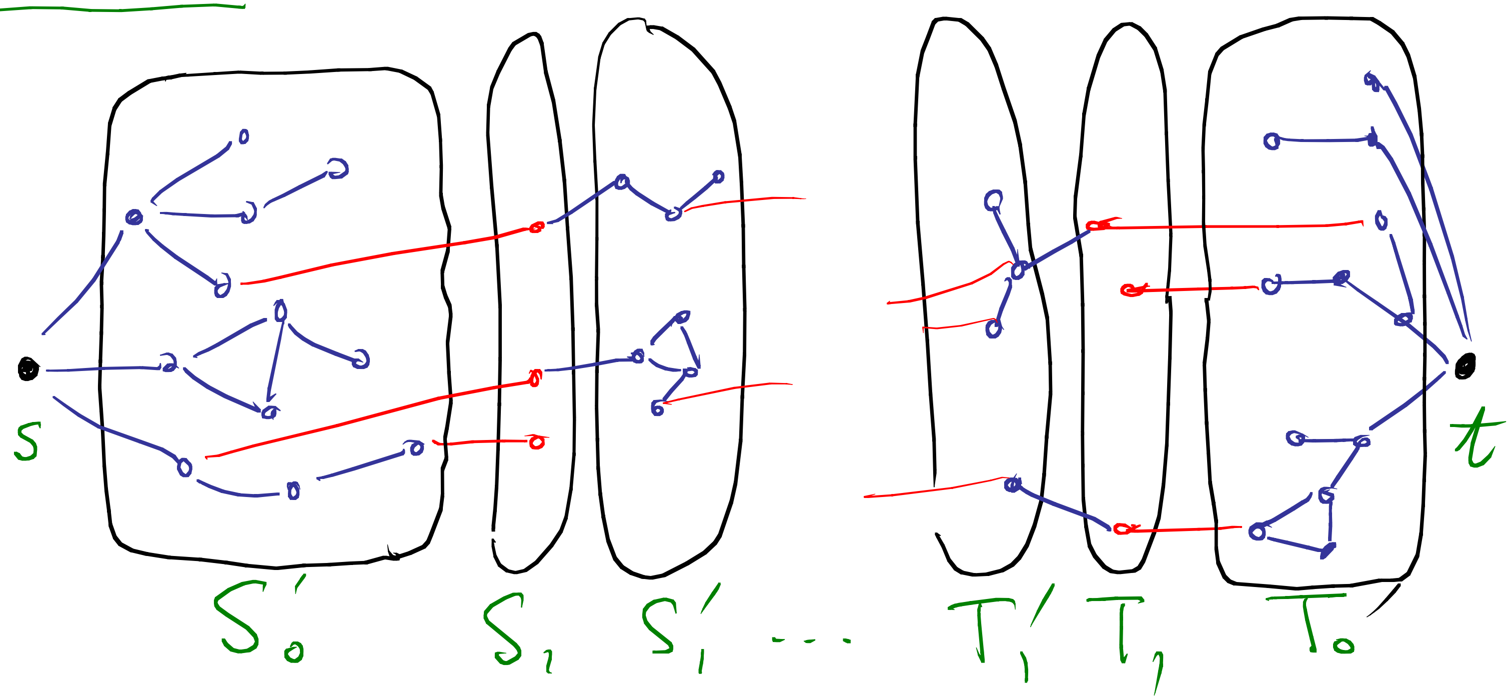
Now, we let S_1', T_1' be defined by

$$S_1' = \{v : d_{\bar{G}}(S_0', v) \leq 5\varepsilon^{-1}\}$$

and let

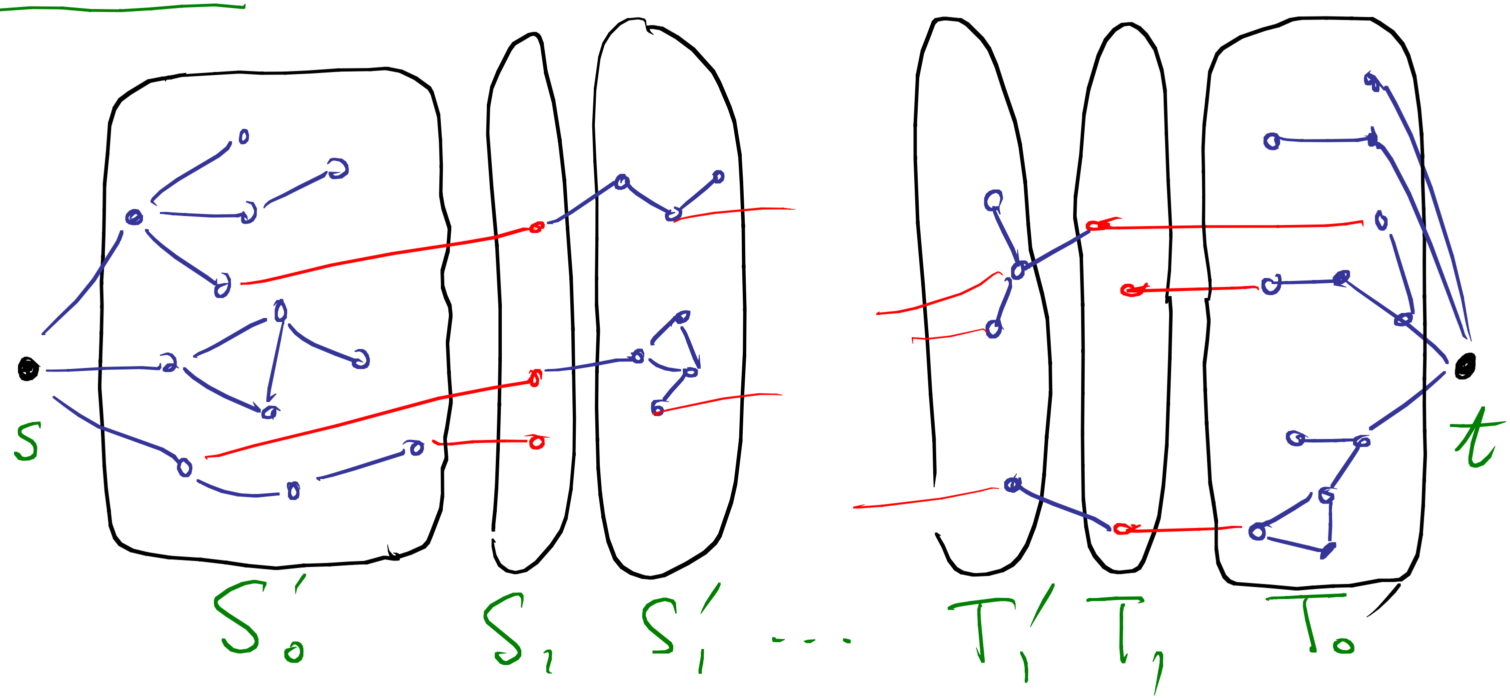
S_2, T_2 be the R-nbrs of of S_1', T_1'.

Proof:



So we have
$$\mathbb{E}[|S_i|] \gtrsim \mathbb{E}[|S'_{i-1}|]\, n \cdot \frac{\varepsilon}{n}$$
$$\geq \left(|S_{i-1}|\, 5\varepsilon^{-1}\right) n \cdot \frac{\varepsilon}{n}$$
$$= 5|S_{i-1}|.$$

Proof:



So we have $\mathbb{E}[|S_i|] \gtrsim \mathbb{E}[|S_{i-1}'|] n \cdot \frac{\varepsilon}{n}$

$$\geq (|S_{i-1}| 5 \varepsilon^{-1}) n \cdot \frac{\varepsilon}{n}$$

$$= 5 |S_{i-1}|.$$

S_i and T_i meet before approx gets too inaccurate.

Application 1: Strong connectivity

Application 1: Strong connectivity

Input: Digraph D

Problem: Accept iff D is strongly connected. (If directed path from any node to any other node.)

Application 1: Strong connectivity

Input: Digraph D

Problem: Accept iff D is strongly connected. (If directed path from any node to any other node.)

- Strong connectivity is NL-complete

Application 1: Strong connectivity

Input: Digraph D

Problem: Accept iff D is strongly connected. (If directed path from any node to any other node.)

- Strong connectivity is NL-complete
- Even when restricted to bounded degree instances.

Application 1: Strong connectivity

- But (for bounded degree instances) a log-space alg succeeds whp on smooth instances.

Application 1: Strong connectivity

- But (for bounded degree instances) a log-space alg succeeds whp on smooth instances.

ALG: Look for short ($O(\log n)$) paths. Accept if you find them between every pair of nodes.

Application 1: Strong connectivity

What does it mean "succeed whp"?

Application 1: Strong connectivity

What does it mean "succeed whp"?

$$D = \bar{D} \oplus R$$

Application 1: Strong connectivity

What does it mean "succeed whp"?

$$D = \bar{D} \oplus R$$

If D is not strong-conn, always reject.

Application 1: Strong connectivity

What does it mean "succeed whp"?

$$D = \bar{D} \oplus R$$

If D is not strong-conn, always reject.

If D is strong-conn, whp accept

Application 1: Strong connectivity

What does it mean "succeed whp"?

$$D = \bar{D} \oplus R$$

If D is not strong-conn, always reject.
If D is strong-conn, whp accept,
(even if $\bar{D}$ is not strong-conn).

Application 1: Strong connectivity

What does it mean "succeed whp"?

$$D = \bar{D} \oplus R$$

If D is not strong-conn, always reject.
If D is strong-conn, whp accept,
(even if $\bar{D}$ is not strong-conn).

Elements of Pf: 1. Add edges $\rightarrow$ low diam
2. degree stays "pretty bounded"
3. If $\bar{D}$ not conn, D becomes conn in a good way

Application 1: Strong connectivity

- What about (s,t)-connectivity?

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.

  Add n^3 dummy nodes.

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.

  Add n^3 dummy nodes.

  Smooth that.

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.

  Add n^3 dummy nodes.

  Smooth that.

  Solve smoothed instance.

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.
  Add n^3 dummy nodes.
  Smooth that.
  Solve smoothed instance.

  Whp, get correct ans for I.

Application 1: Strong connectivity

- What about (s,t)-connectivity?

  A similar theorem is "impossible".

- If an alg succeeds whp on smoothed instances, can do this:

  Take an instance I on n nodes.
  Add n^3 dummy nodes.
  Smooth that.
  Solve smoothed instance.

  Whp, get correct ans for I.

  (More complicated than it sounds in log-space)

Application 2:

- A graph is **k-linked** if for all terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ there are k disjoint paths from s_i to t_i.

Application 2:

- A graph is **k-linked** if for all terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ there are k disjoint paths from s_i to t_i.
- Recognizing k-linked graphs is NP-hard for $k \geq 2$.

Application 2:

- A graph is **k-linked** if for all terminal pairs $(s_1, t_1), \dots, (s_k, t_k)$ there are k disjoint paths from s_i to t_i.
- Recognizing k-linked graphs is NP-hard for $k \geq 2$.
- Want alg for smoothed instances as in previous application.

Application 2:

- A graph is k-linked if for all terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ there are k disjoint paths from s_i to t_i.
- Recognizing k-linked graphs is NP-hard for $k \geq 2$.
- Want alg for smoothed instances as in previous application. (Doesn't work yet...)

Application 2:

- Difficulty: Can't rule out possibility: $\bar{D}$ is not k-linked but D is k-linked in a way not witnessed by short paths.

Application 2: Property testing

- Difficulty: Can't rule out possibility: $\bar{D}$ is not k-linked but D is k-linked in a way not witnessed by short paths.

Application 2: Property testing

- Difficulty: Can't rule out possibility: $\bar{D}$ is not k-linked but D is k-linked in a way not witnessed by short paths.

- This case doesn't come up in property testing: TESTING k-linked:
  - If $\bar{D}$ is k-linked accept
  - If $\bar{D}$ is ε-far from k-linked reject
  - o.w. whatever

Application 2: Property testing

Alg: (for testing k-linked)

a

Application 2: Property testing

Alg: (for testing k-linked)

Given a bounded degree digraph $\overline{D}$,

Application 2: Property testing

Alg: (for testing k-linked)

Given a bounded degree digraph $\overline{D}$,

For each node, with pr $\frac{\varepsilon}{2}$ add an

arc out to some node chousen u.a.r.

Application 2: Property testing

Alg: (for testing k-linked)

Given a bounded degree digraph $\bar{D}$,

For each node, with pr $\frac{\varepsilon}{2}$ add an

arc out to some node chosen u.a.r.

Search all terminal sets for disjoint

paths of length $O(\log n)$.

Application 2: Property testing

Alg: (for testing k-linked)

Given a bounded degree digraph $\bar{D}$,

For each node, with pr $\frac{\varepsilon}{2}$ add an arc out to some node chosen u.a.r.

Search all terminal sets for disjoint paths of length $O(\log n)$.

If $\bar{D}$ is k-linked then D is also, and whp we accept. If $\bar{D}$ is ε-far, then D is not k-linked $\Rightarrow$ reject.

Conclusion:

- Add εn random edges to any connected graph, get diameter $O(\varepsilon^{-1}\log n)$ whp.

Conclusion:

- Add εn random edges to any connected graph, get diameter $O(\varepsilon^{-1}\log n)$ whp.

So:

Conclusion:

- Add εn random edges to any connected graph, get diameter $O(\varepsilon^{-1}\log n)$ whp.

So:

- Smoothed instances of strong-conn are recognized by log-space alg whp (analogous claim should not hold for (s,t)-conn)

Conclusion:

- Add εn random edges to any connected graph, get diameter $O(\varepsilon^{-1}\log n)$ whp.

So:

- Smoothed instances of strong-conn are recognized by log-space alg whp (analogous claim should not hold for (s,t)-conn)

- k-linkedness property is testable in poly-time.