How random walks led to advances in testing minor-freeness

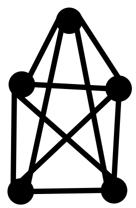
C. Seshadhri
(UC Santa Cruz)

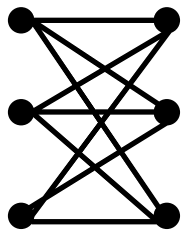
My coauthors

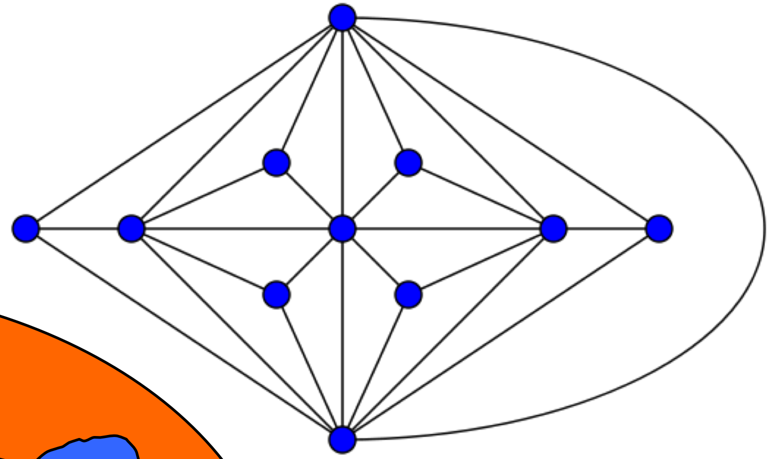
Akash Kumar, Purdue

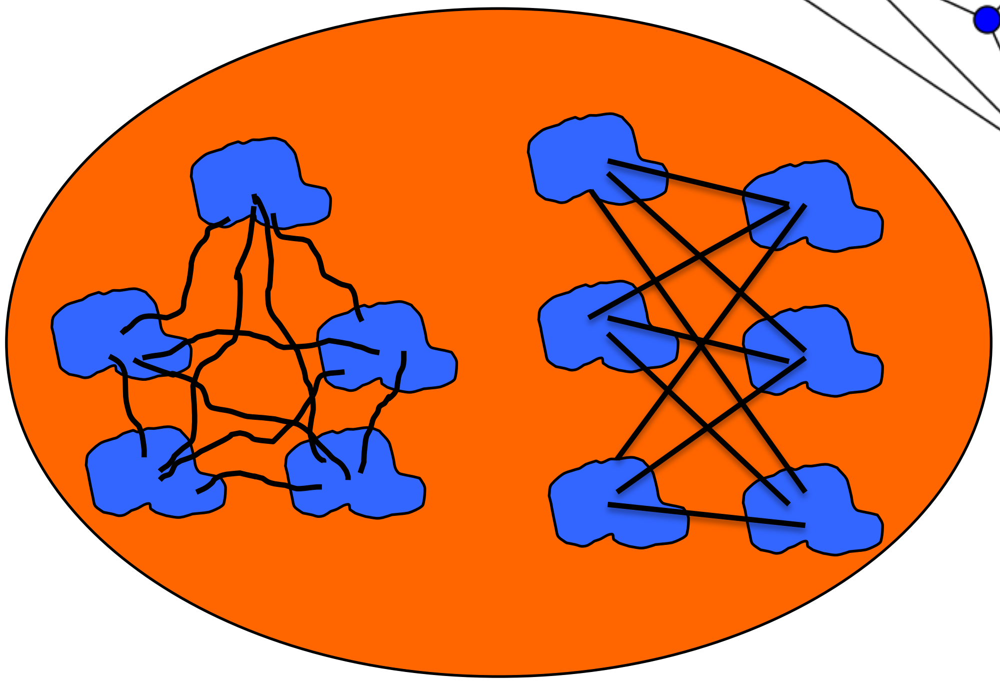
Andrew Stolman, UCSC

Classics









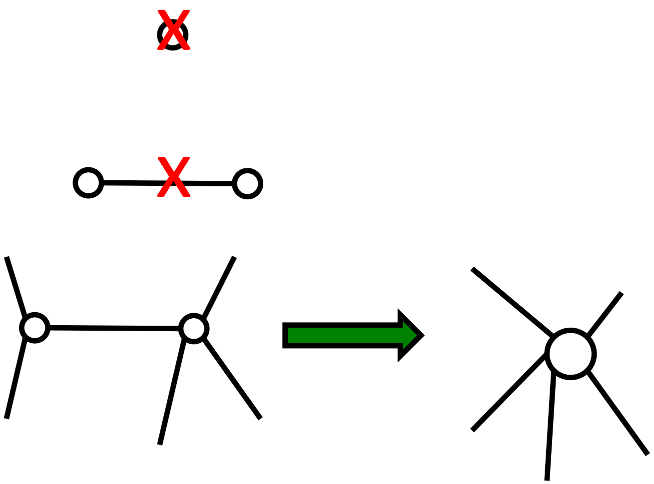
- [Kuratowski 30, Wagner 37]

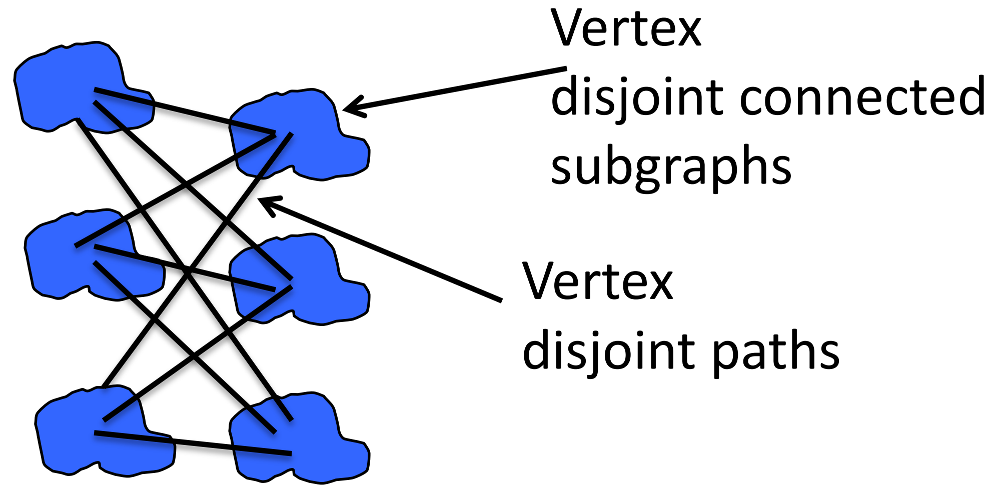
G is not planar, iff it contains a K_5 or $K_{3,3}$ minor

- From geometry to topology

https://en.wikipedia.org/wiki/Planar_graph#/media/File:Goldner-Harary_graph.svg

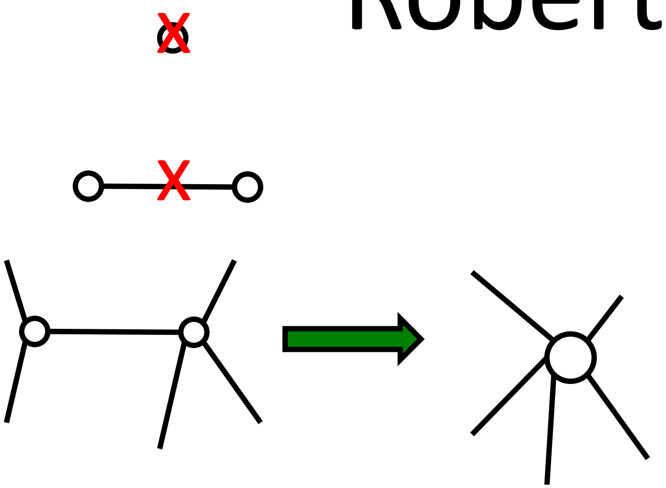
Minors

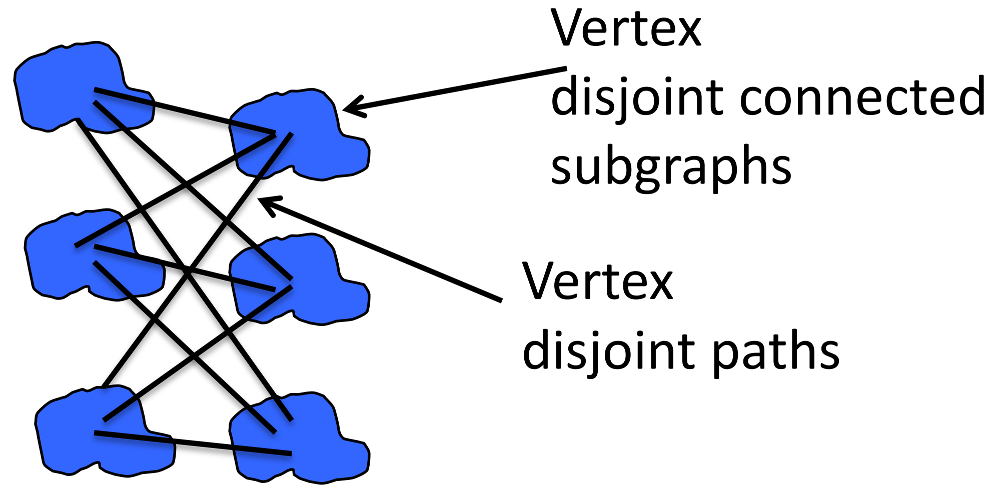




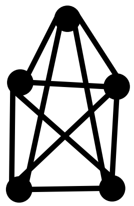
- H is minor of G, if H obtained by deletions and edge contractions in G
- Forbidden minor characterization: G is planar iff it does not contain K_5 and $K_{3,3}$ minors
  - G is forest, iff it doesn’t have K_3 minor

Robertson-Seymour I - XX

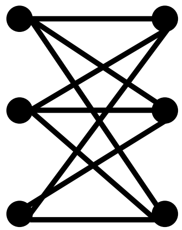


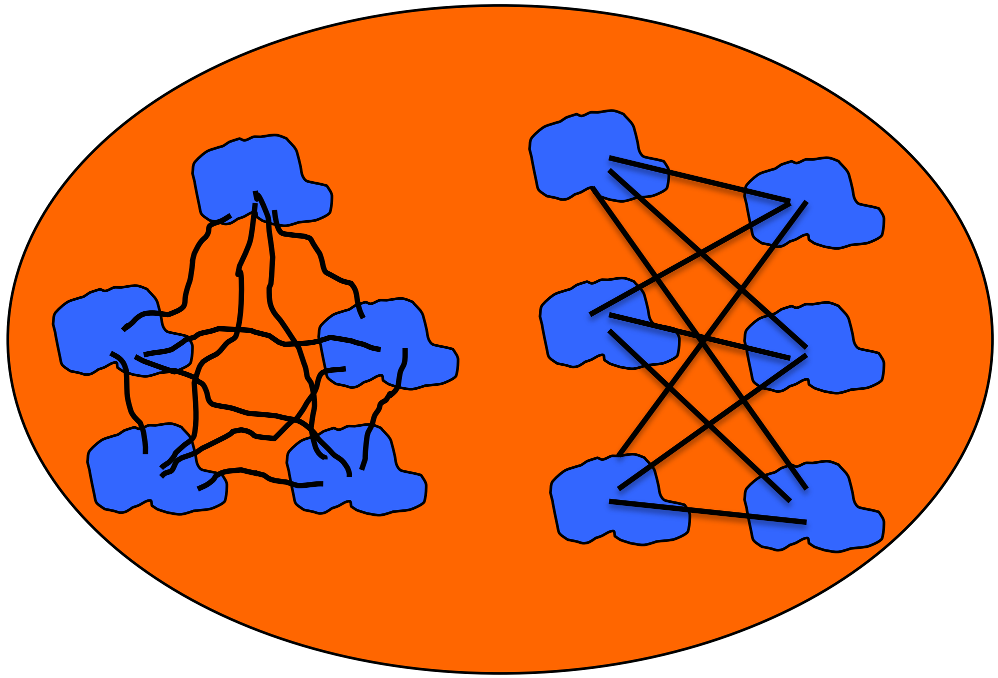


- If property P is closed on taking minors, P has finite forbidden minor characterization
- Planarity, outerplanarity, bounded genus embeddable, treewidth < k,...
  - Each P has a finite list F of forbidden minors



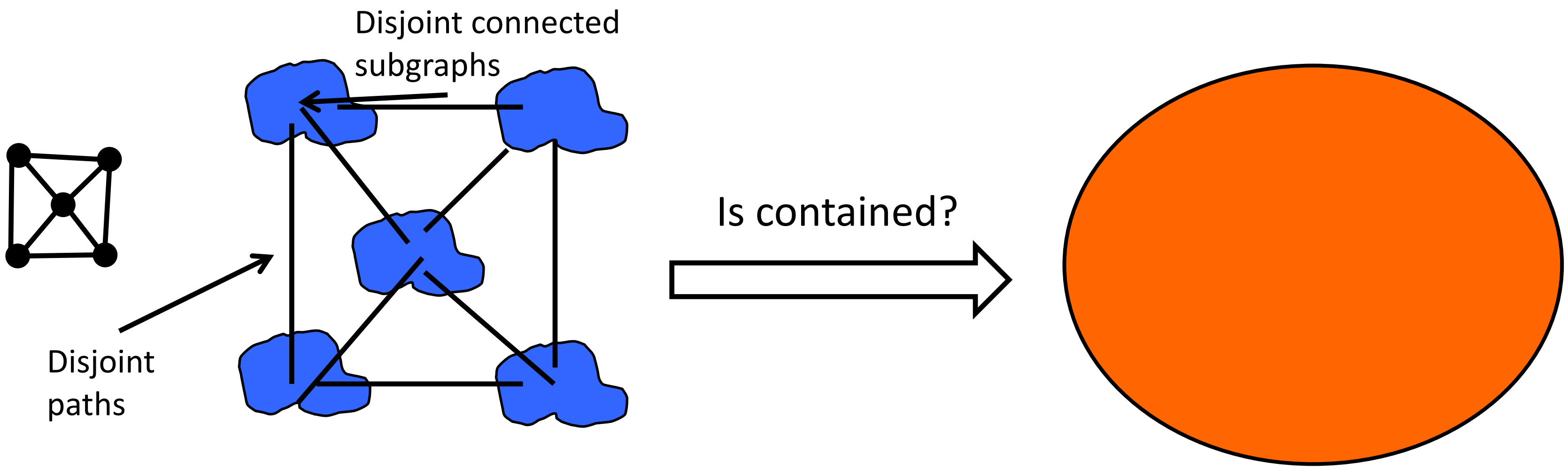
Algorithmic classics





- Given non-planar G, find forbidden minor in it
- [Hopcroft-Tarjan 74] O(n) time algorithm to decide planarity

Robertson-Seymour: algorithms

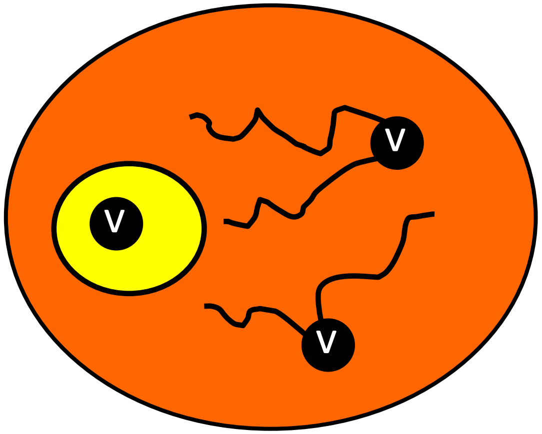


- There is $O(n^3)$ algorithm to decide if G contains H-minor
  - Thus, $O(n^3)$ for any minor-closed property
- [Kawarabayashi-Kobayashi-Reed12] $O(n^2)$ algorithm
- Grand generalization of Hopcroft-Tarjan, worse running time

What if you can’t read all of G?

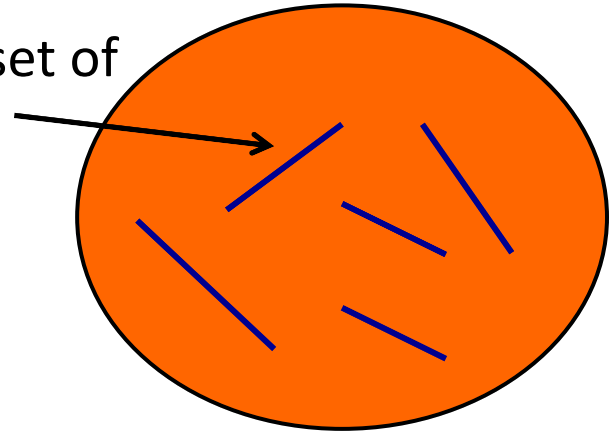
o(n) algorithms for planarity

[Goldreich-Ron 02] The query model



- G is bounded degree, stored as adjacency list
  - n vertices, d degree bound
- You can pick random vertices/seeds
- You can crawl from these seeds
  - BFS, Random walks

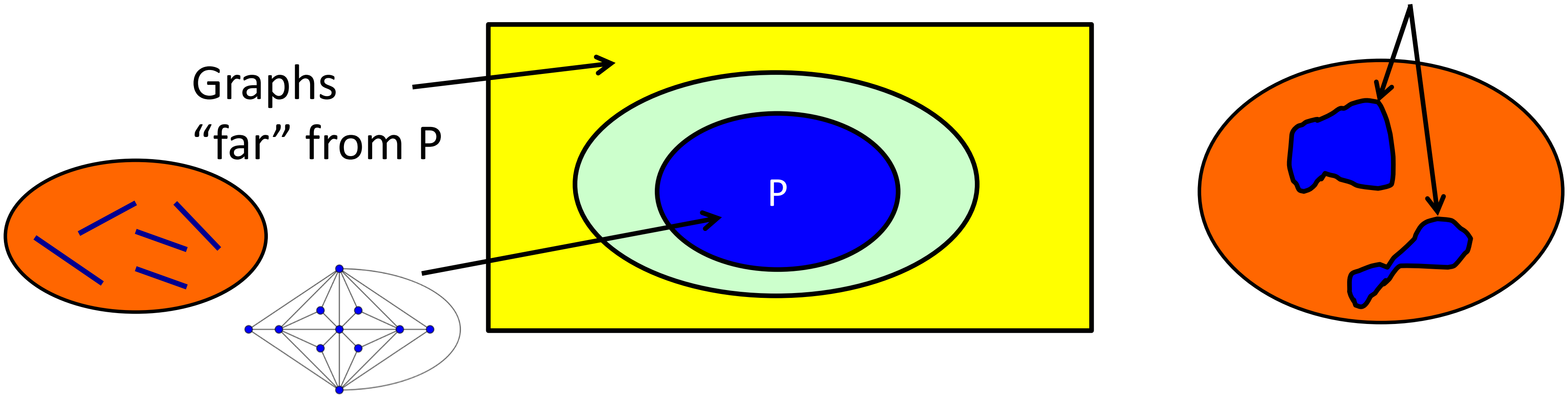
Distance to planarity



- G is ε-far from planar if > εnd edges need to be removed to make G planar

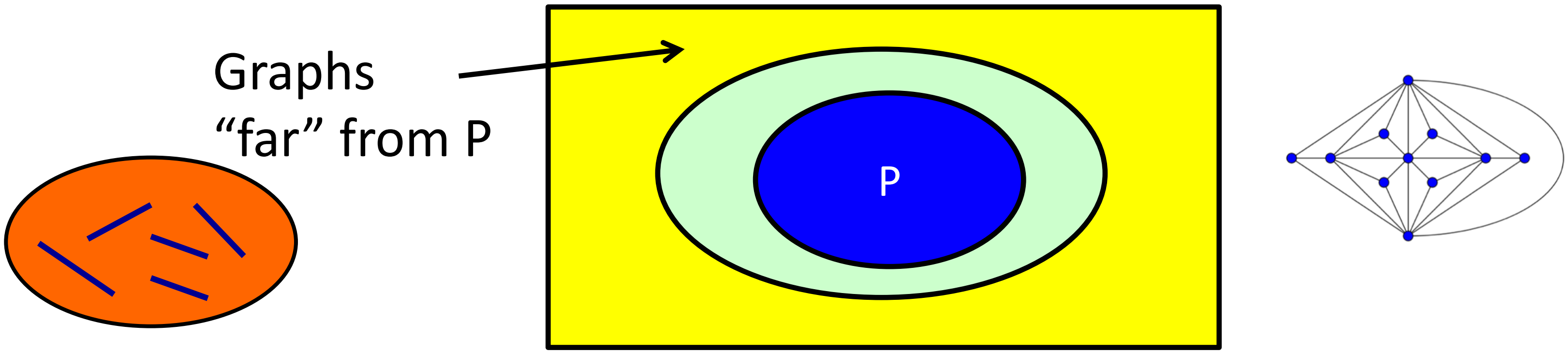
- G is ε-far from H-minor freeness if > εnd edges need to be removed to make H-minor free

The testing problem



- If G is ε-far from planar, reject w.p. > 2/3
- (Two-sided) If G is planar, accept w.p. > 2/3
- (One-sided) If G is planar, accept w.p. 1
- (One-sided) If G is ε-far from planar, find forbidden minor w.p. > 2/3

[Benjamini-Schramm-Shapira 08]



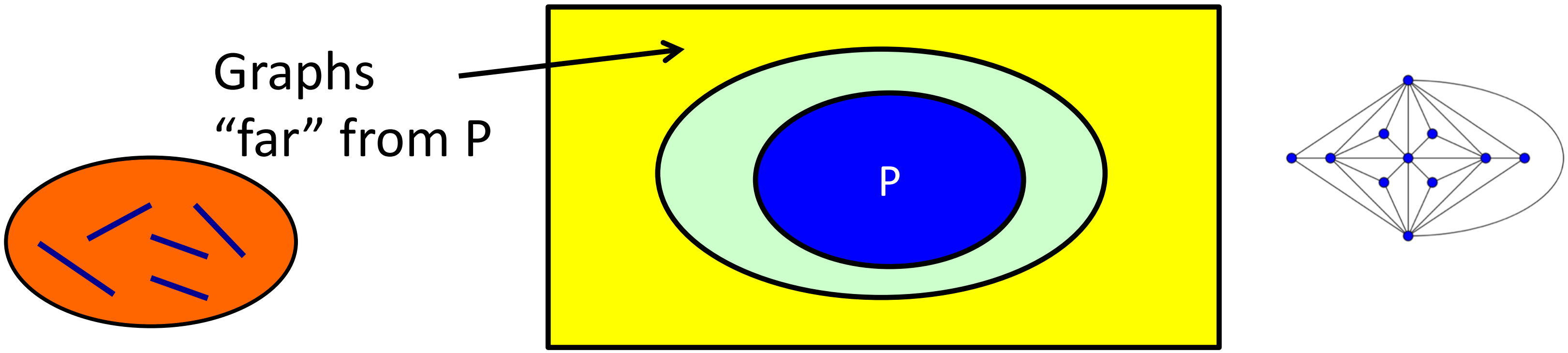
- Two-sided tester for all minor-closed properties in exp(exp(exp(d/ε)) queries

- [Goldreich-Ron 02, Czumaj-Goldreich-Ron-S-Shapira-Sohler 14]

One-sided $\sqrt{n}$ lower bound

  – Forbidden minor is poly(log n) sized

Post BSS08



Two-sided

- [Hassidim-Kelner-Nguyen-Onak 09]
  exp(d/ε)
- [Levi-Ron 15] $(d/\varepsilon)^{\log(1/\varepsilon)}$
- [Yoshida-Ito 11, Edelman-Hassidim-Nguyen-Onak 11]

  poly(d/ε) for bounded treewidth classes

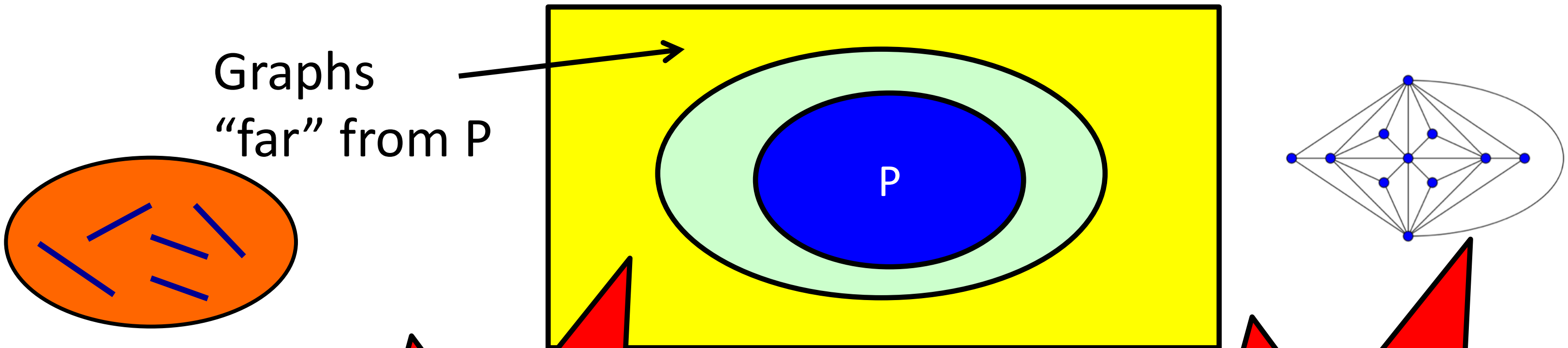
One-sided

- [Czumaj-Goldreich-Ron-S-Shapira-Sohler 14]

  $\sqrt{n}$ for cycle-freeness
- [Fichtenburger-Levi-Vasudev-Wotzel17]

  $n^{2/3}$ for $K_{2,r}$-minor freeness

Post BSS08



Tw

One-s

- [Hass
- [L

dim-

[Czur

tzel17]

ded

sses

fr

poly(d/ε) tester for planarity?

$\sqrt{n}$ one-sided tester for planarity?

Sorry, this is a marketing slide

BSS08

H-minor freeness with $o(\log n)$ queries and one-sided error. In fact, a much stronger $\Omega(\sqrt{n})$ lower bound can be deduced by adapting an argument from [22]. We raise the following conjecture, stating that the $\Omega(\sqrt{n})$ lower bound is tight.

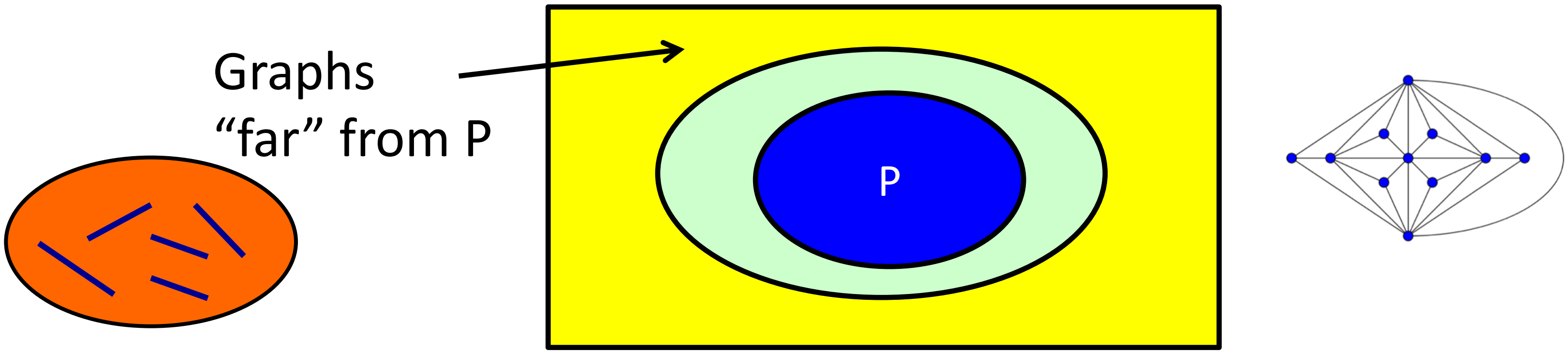
Conjecture 5.2. *For every H, being H-minor free can be tested in the bounded degree setting with one-sided error and query complexity $\tilde{O}(\sqrt{n})$.*

If the conjecture is true, then the Graph Minor Theorem [34] implies that the same is true for any minor-closed graph property.

Open Problem 9.26 (improving the upper bound of Theorem 9.25): *Can any minor-closed property be tested in query* (and time) *complexity that is polynomial in d/ϵ? What about the special case of* `Planarity`*?*

Goldreich17

And now…



Two-sided

[Kumar-S-Stolman 19]

poly(d/ε) for all minor-closed properties

One-sided

[Kumar-S-Stolman 18]

$d\sqrt{n} \cdot n^{o(1)}$ for all minor-closed properties

Based on (new?) toolkit using spectral graph theory for minor-freeness

One-sided tester

Planarity, outerplanarity, series-parallel, bounded genus embeddable, treewidth < k

[Kumar-S-Stolman 18]

Fix minor-closed property P. (By [RS], there is finite list of forbidden minors.)

There is $O^*(d\sqrt{n})$- time randomized algorithm:

If G is ε-far from P, algorithm produces a forbidden minor in G

- O*() hides poly(1/ε).$n^{o(1)}$
- Doubly exponential dependence on r, size of largest minor in G

Two-sided tester

[Kumar-S-Stolman 19]

Fix minor-closed property P.

There is $O(d\varepsilon^{-100})$ time two-sided tester for P

– Previously, poly(1/ε) not known for planarity

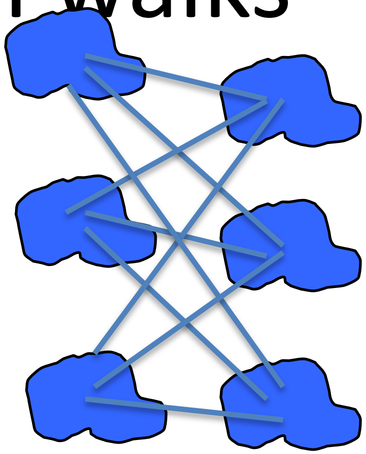
Cute corollary

Consider d bounded degree G with at least $(3+\varepsilon)n$ edges.

There is $O^*(dn^{1/2})$-time algorithm that finds K_5 or $K_{3,3}$ minor in G

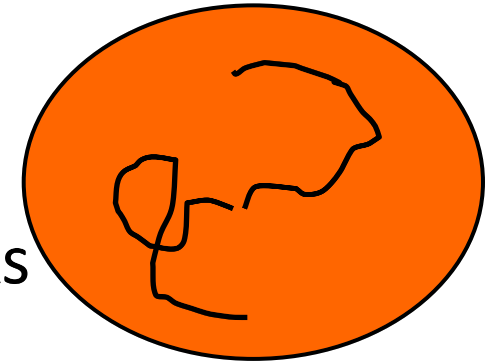
- Analogous theorem for any minor-closed property

Less graph minors, more random walks



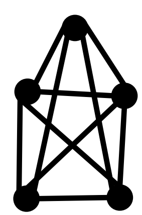
- No Robertson-Seymour machinery
  - No brambles, treewidth, etc.
  - In searching for H-minor, H does not play major role

- It's all spectral graph theory
  - Finding minors through random walks

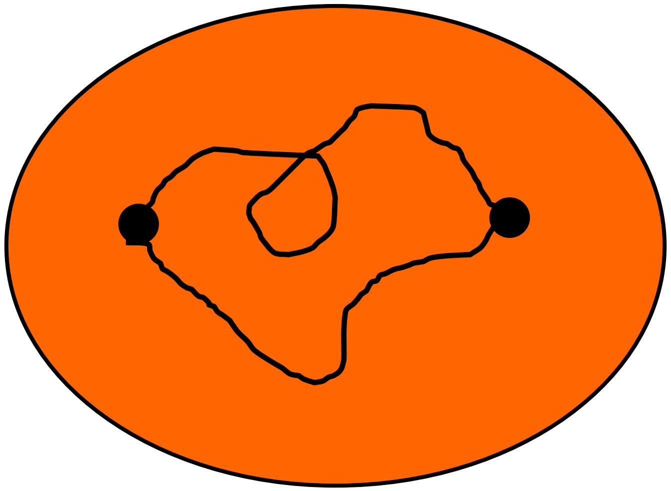


How did it all start?

Let's try to find K_5 minors

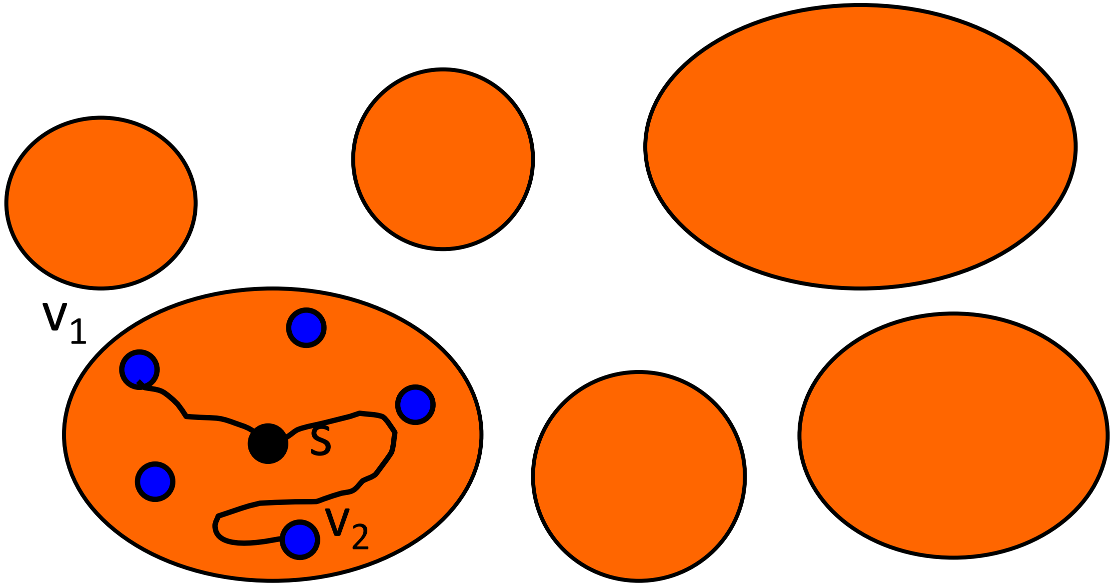


[Goldreich-Ron 99]



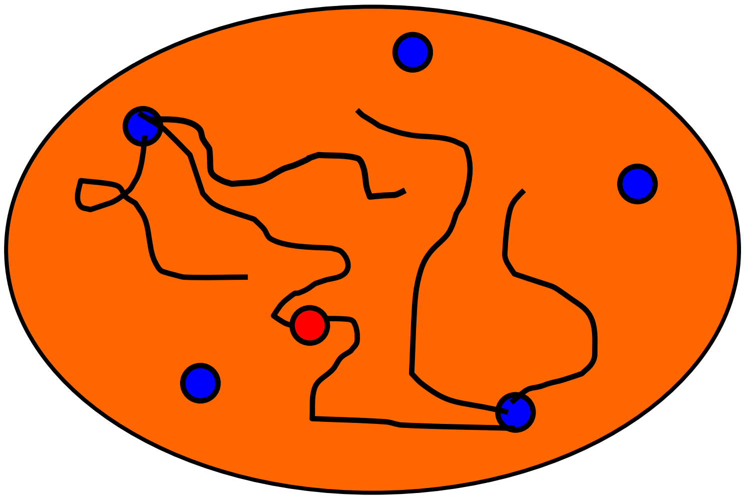
- If G is ε-far from bipartite, $\sqrt{n}$ algorithm to find odd cycle
  - – The inspiration for our result
  - – Finding cycles through random walks

The rapid mixing case: G is expander



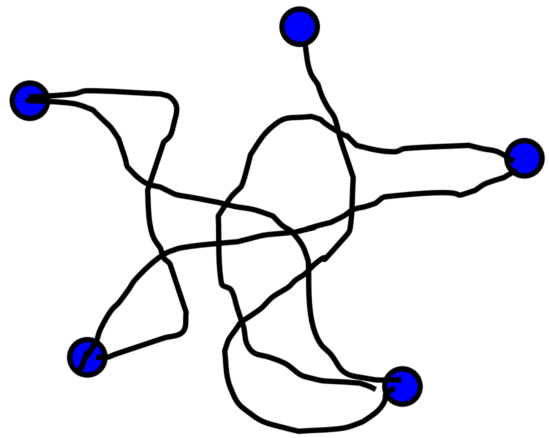
- G is disjoint collection of expanders
  - ℓ = log n
- Pick random starting vertex s
- Perform 5 ℓ-length rws from s to reach v_1, v_2,…, v_5
  - Perform $\sqrt{n}$ random walks from v_1…v_5 to form K_5 minor

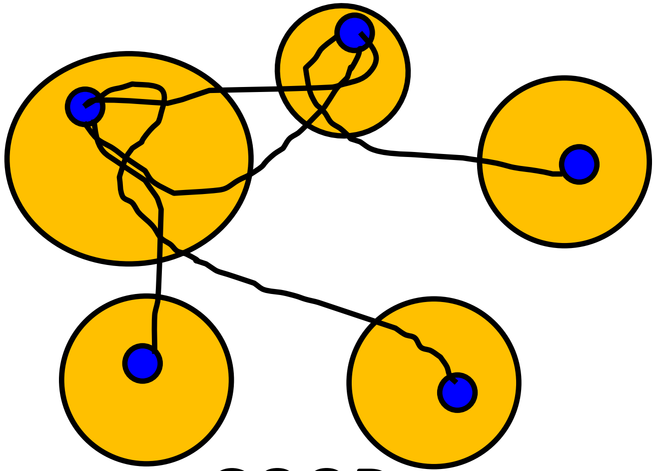
Connecting the dots



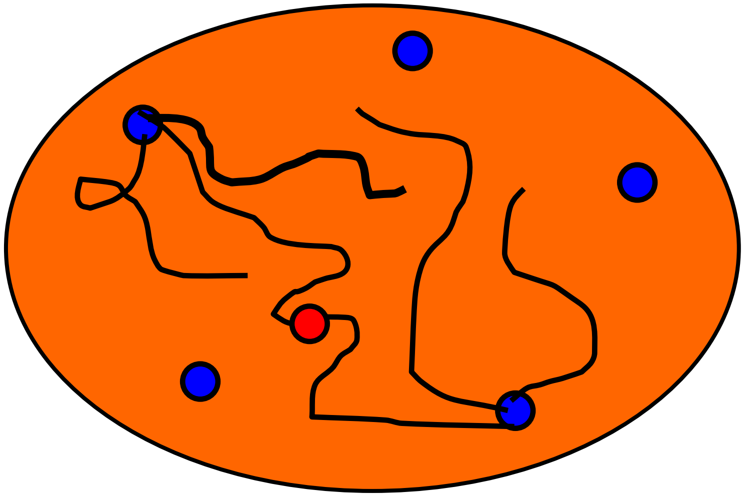
- Perform $\sqrt{n}$ ℓ-length random walks from v_i
  - Birthday paradox: guaranteed to have two walks end at the same vertex
- Guaranteed to connect all (v_i, v_j) pairs
  - Union bound

Paths don't imply minors



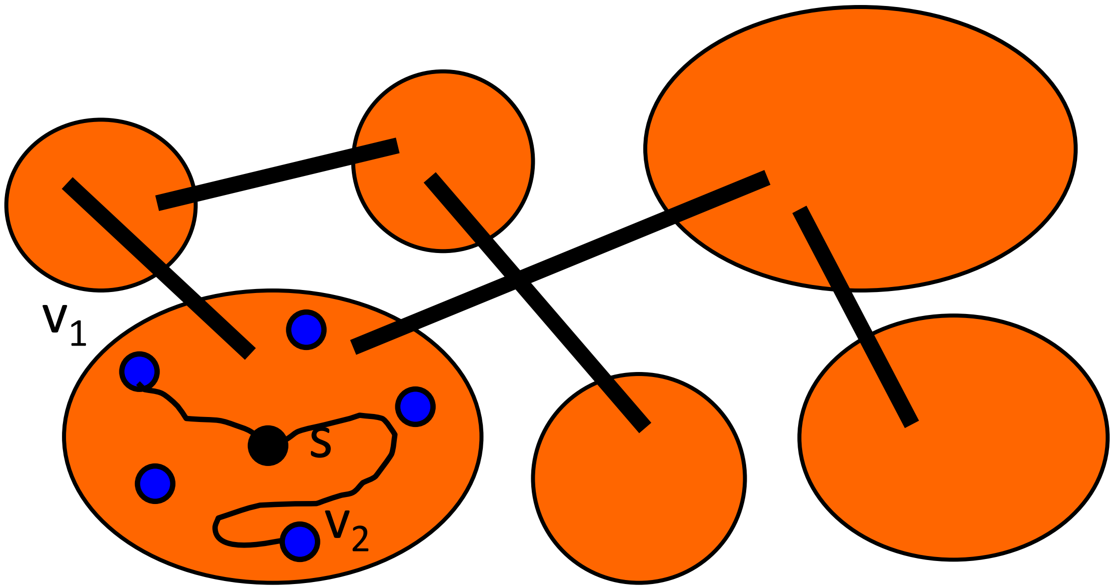


- Paths unlikely to be (internally) vertex disjoint
- In expander, intersections are "localized"
  - We can contract away intersections to get K_5



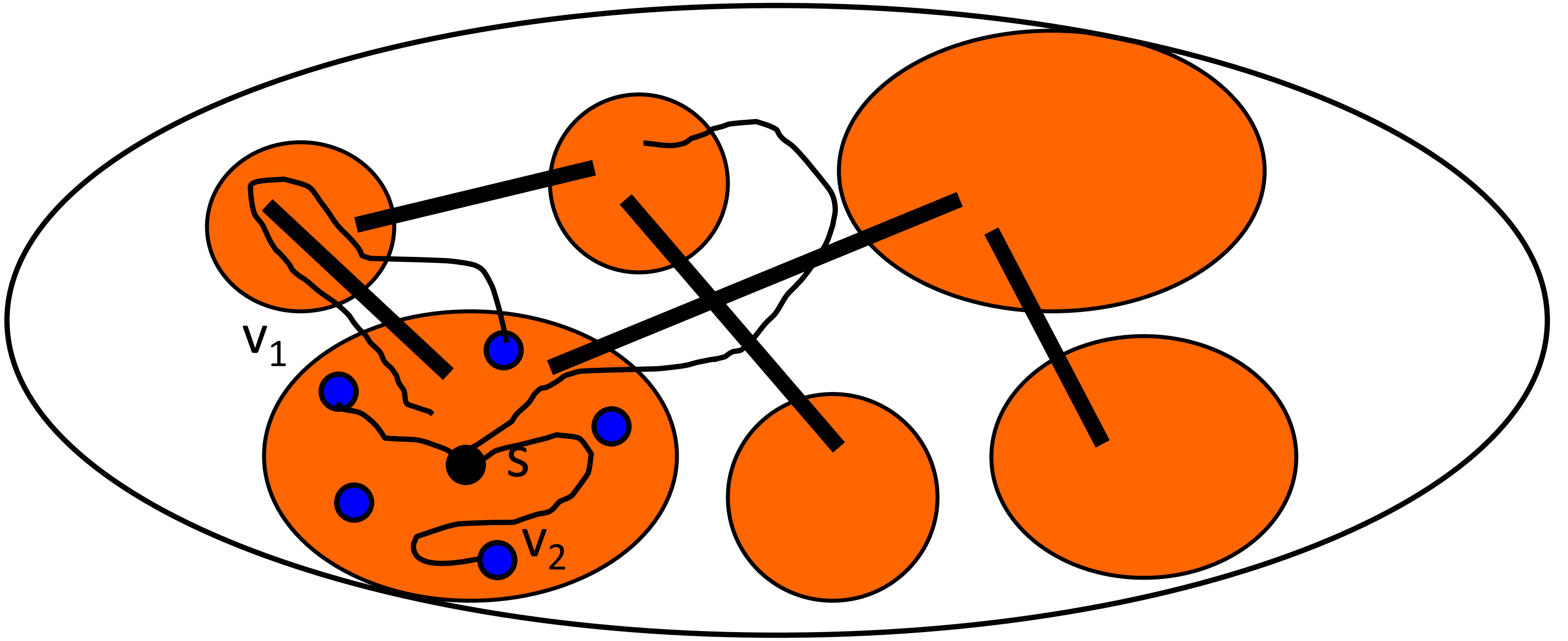
Just run this algorithm on any graph?

[GR99] The general case



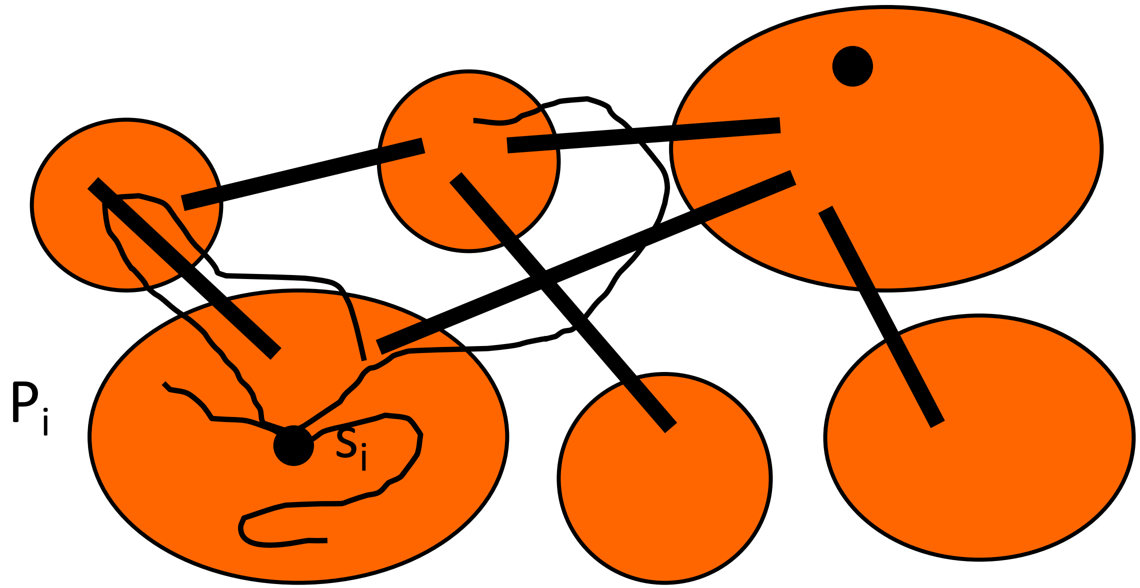
- Every graph can be decomposed into "expander-like" pieces
  - Remove εdn edges, get disjoint pieces with mixing time poly(log n)
- [Trevisan 05, Arora-Barak-Steurer 15] Deep connection with UGC/approx algorithms

The sublinear constraint



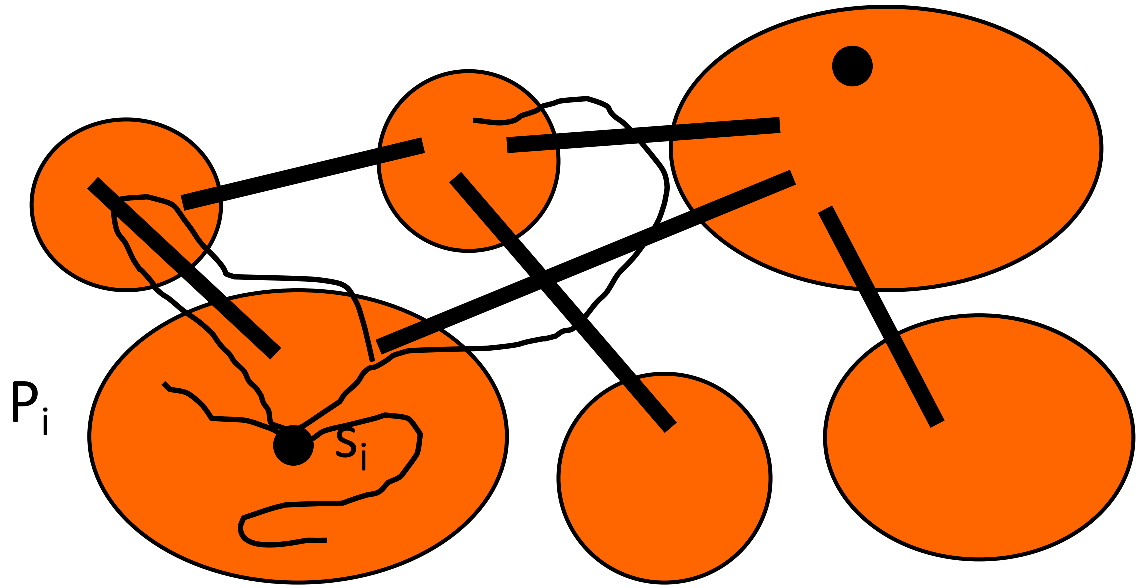
- G can be decomposed into G’, disjoint collection of “expander-like” pieces
- Yes, but o(n) algorithm cannot know G’
- Algorithm performs random walks on G, and hopes to simulate expander algorithm on G’…?

The [GR99] decomposition



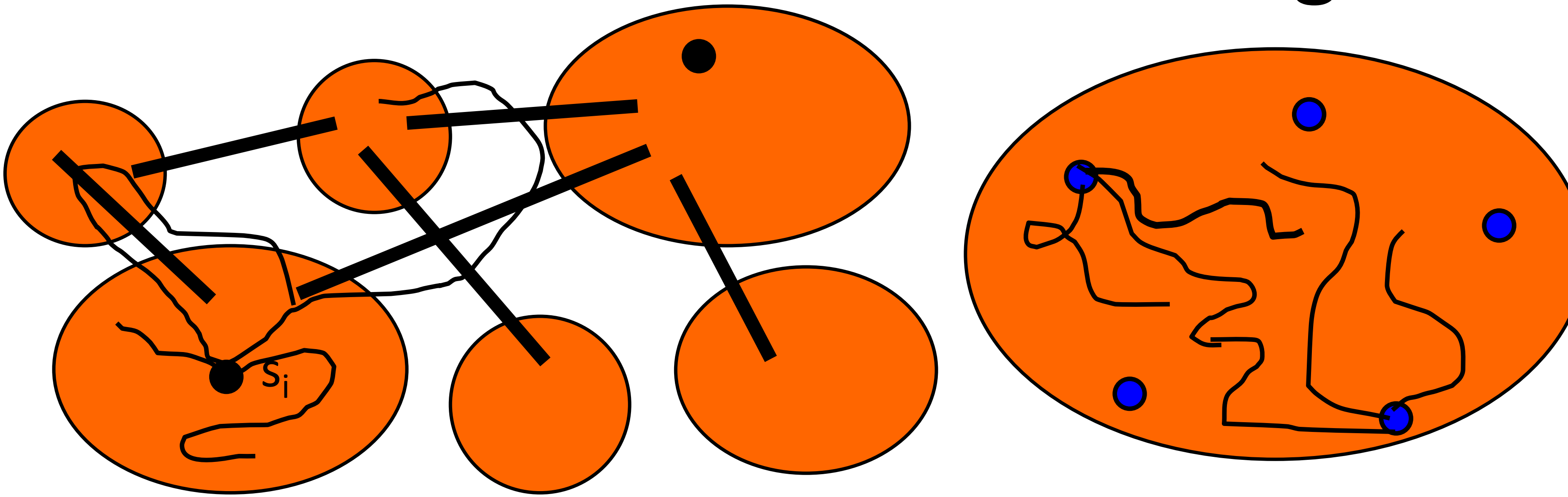
- (There is k st) Pick s_1, s_2, …, s_k uar
- We can remove εdn edges and get pieces P_1, P_2,…P_k where:
- ℓ-rws from s_i (in G) reach all vertices in P_i with roughly the same probability ($> 1/n^{1/2}$)

The [GR99] decomposition



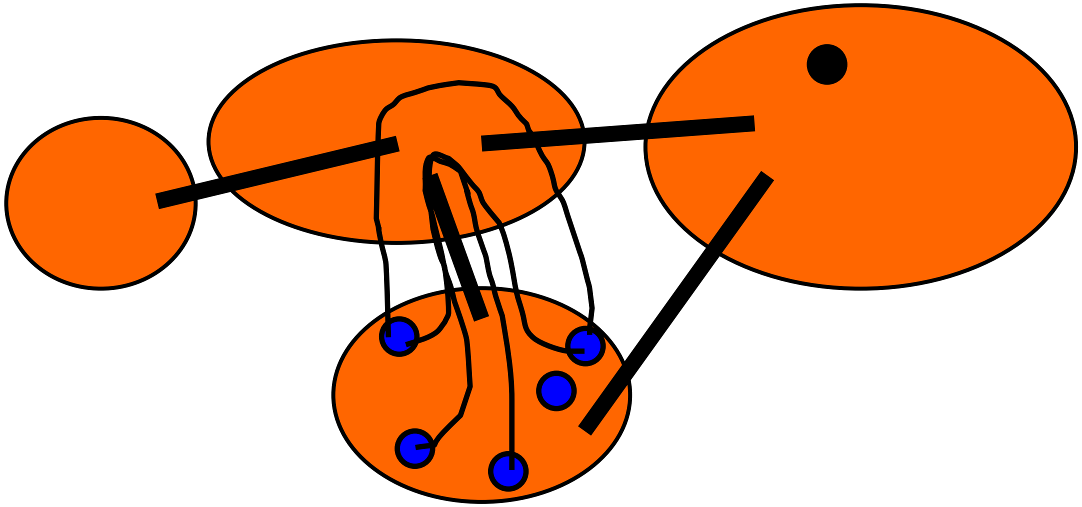
- ℓ-rws from s_i (in G) reach all vertices in P_i with roughly the same probability
- The expander analysis goes through
  - If G is far from bipartite, then constant fraction (by total size) of P_i are far from bipartite

Problem #1 for minor finding



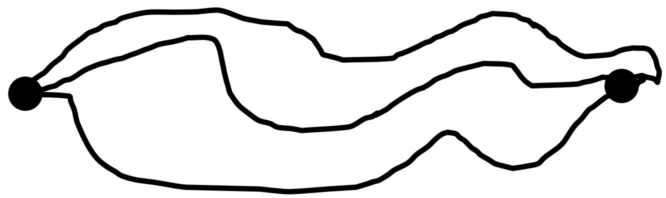
- ℓ-rws from s_i (in G) reach all vertices in P_i with roughly the same probability
  - Only have guarantee from **one vertex** in P_i
  - Enough for finding cycle
- K_5 needs walks from 5 "starting" vertices

Problem #2 for minor finding



- ℓ-rws from s_i (in G) reach all vertices in P_i with roughly the same probability
- These walks leave P_i, and we have no control on intersection
  - No problem for odd-cycle
- How to argue about minors?

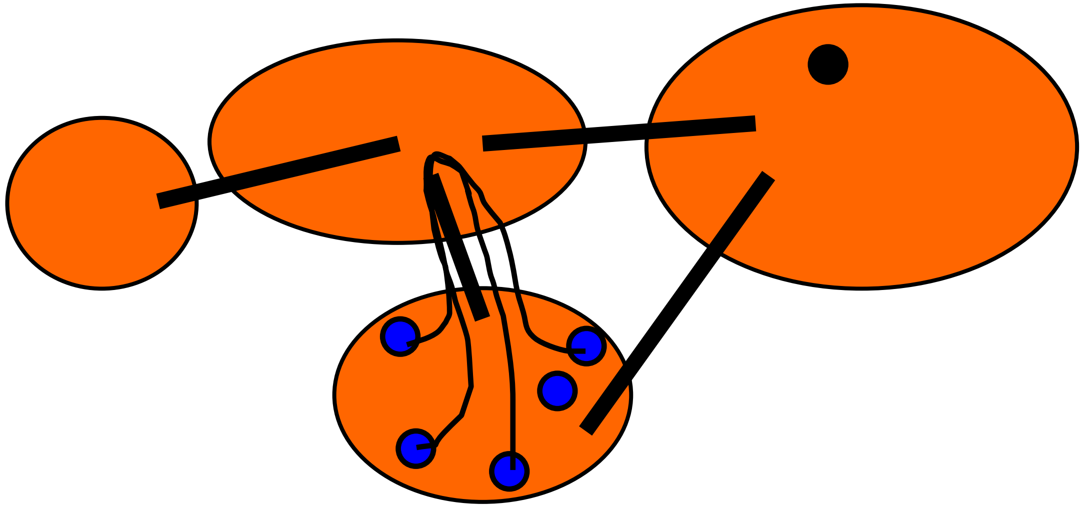
Fixed source and destination



- [Czumaj-Goldreich-Ron-S-Shapira-Sohler 14]

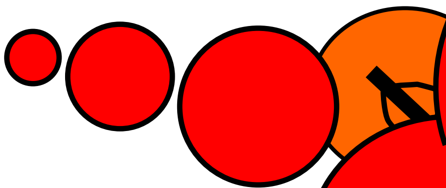
  $\sqrt{n}$ tester H-minor freees, when H is cycle
- [Fichtenburger-Levi-Vasudev-Wotzel17] $n^{2/3}$ algorithm if H is $K_{2,r}$ or cactus graph
- All about finding multiple paths between the same two vertices

Fundamental problem



- For any decomposition…
- Need to walk ℓ > (log n) steps to reach most vertices in each piece
  - There could be εn cut edges
- So walks will leave piece whp, and we don't know how to control the behavior outside

The [GR99] decomposition

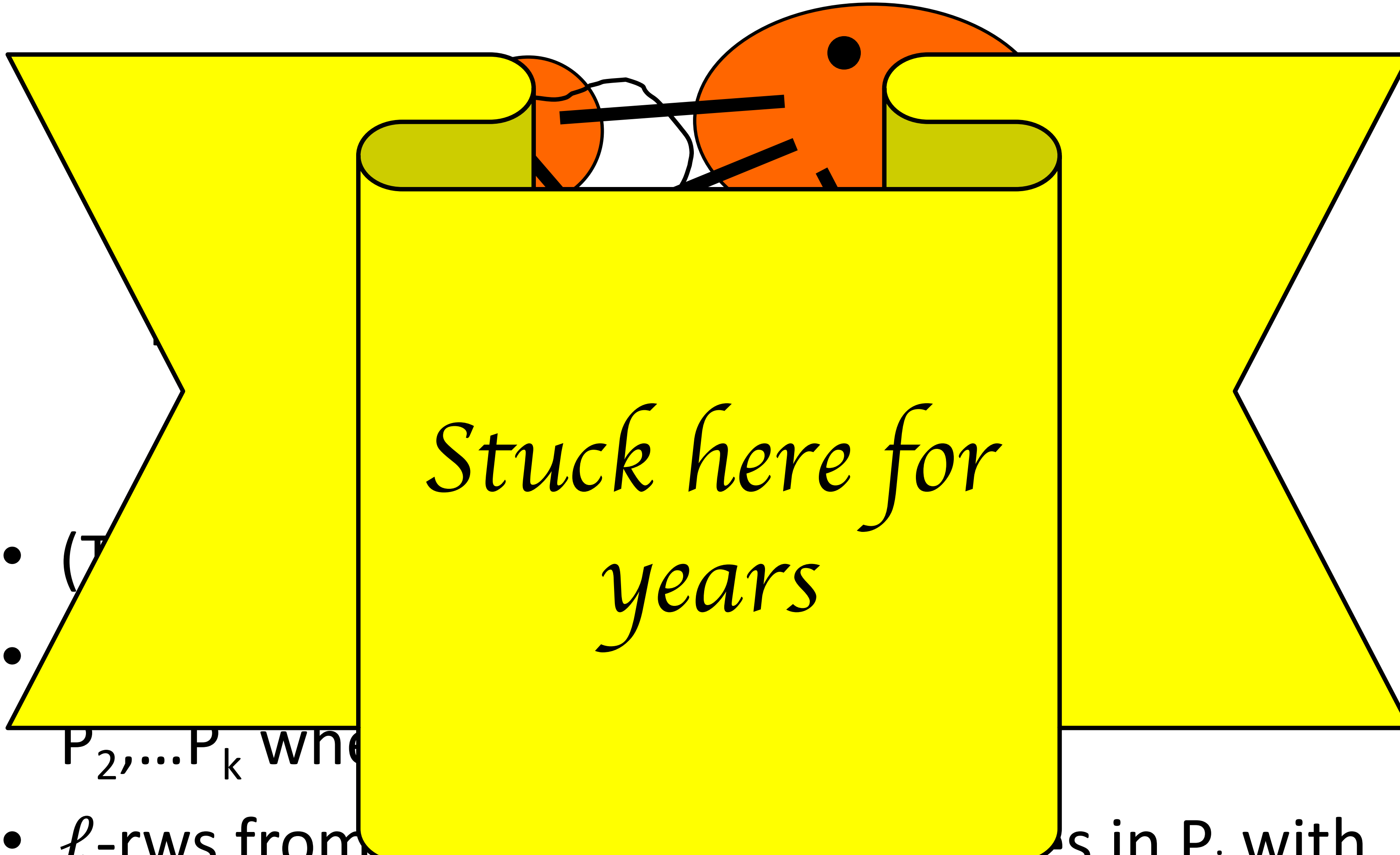


Somehow strengthen this decomposition?

More starting vertices within in each piece?

- (There is
- We can remove εdn … d get pieces P_1, P_2,…P_k where:
- ℓ-rws from s_i (in G) reach all vertices in P_i with roughly the same probability (> $1/n^{1/2}$)

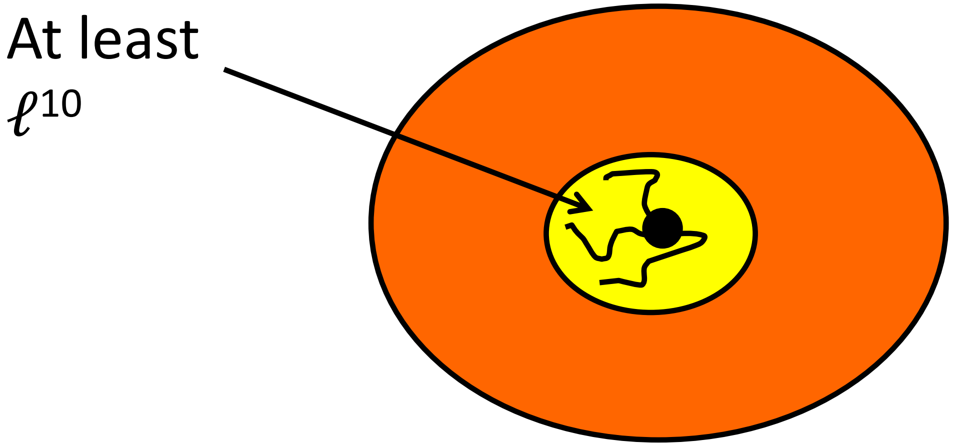
The [GR99] decomposition



- (T
-
  P_2,…P_k whe
- ℓ-rws from … es in P_i with roughly the same probability (> $1/n^{1/2}$)

Revisit the expander case: When can random walks find minors?

Leaking random walks



- $\ell = n^{\delta}$ (think little more than poly(log n))

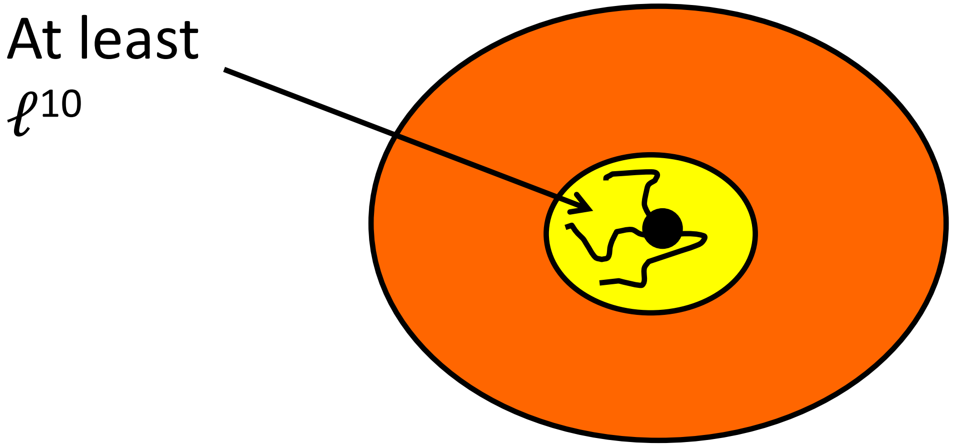
  $\mathbf{p}_{s,\ell}$ = Prob. vector of ℓ rw from s

- s is "leaky" if:

$$\|\mathbf{p}_{s,\ell}\|_2^2 \leq \ell^{-10}$$

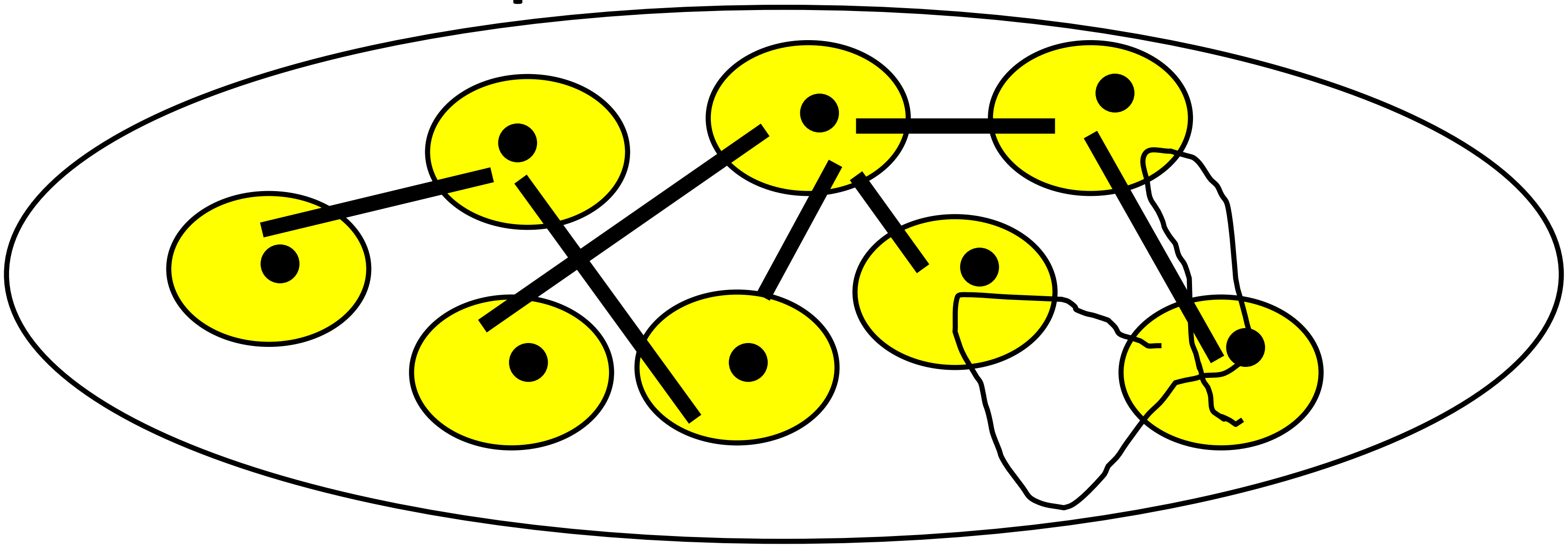
- It means: ℓ-rws from s reach at least poly(ℓ) vertices

The beating heart of one-sided testing



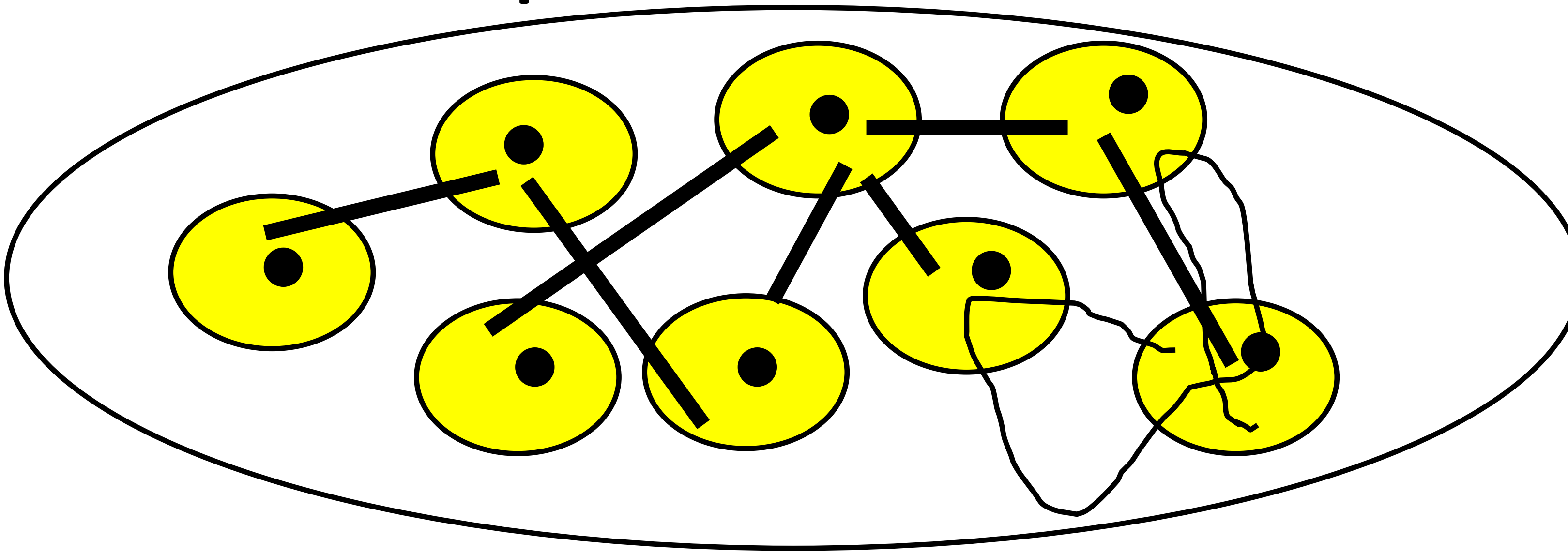
- If there are at least n/ℓ leaky vertices, the random walk algorithm finds K_5 minor whp
  - One doesn't need "expanding" random walks to get algorithm to work
  - For K_r minor-freeness, change polynomial in leaky definition

A decomposition statement



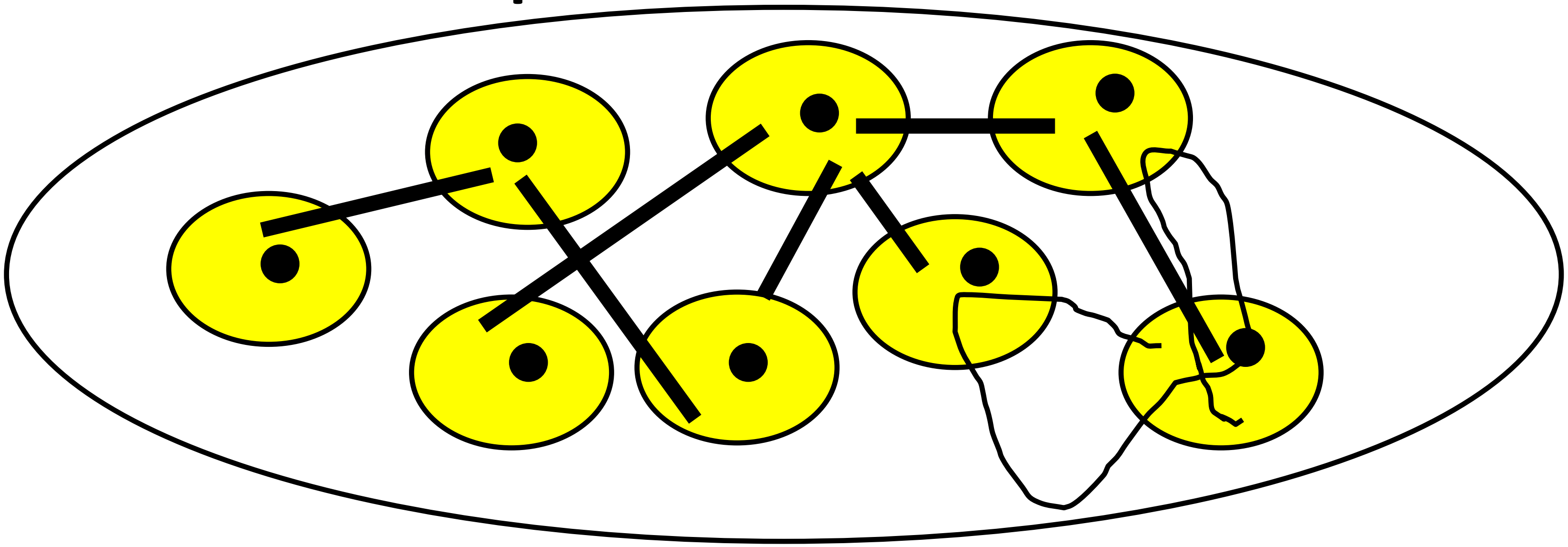
- Suppose there are $< n/\ell$ leaky vertices
  - Rws from most s are “badly” trapped
- Pick $s_1, s_2, \ldots, s_k$ uar
- We can remove εdn edges to get pieces $P_1, P_2 \ldots P_k$ such that:
- Each $|P_i| = \text{poly}(\ell)$ and rws from s_i reach every vertex with P_i with prob $> 1/\text{poly}(\ell)$

A decomposition statement



- Each $|P_i|$ = poly(ℓ) and rws from s_i reach every vertex with P_i with prob > 1/poly(ℓ)
  - poly(ℓ) walks from s_i find superset of P_i
- If G far from planar, many P_is non-planar
  - Find superset of P_i, and run exact algorithm

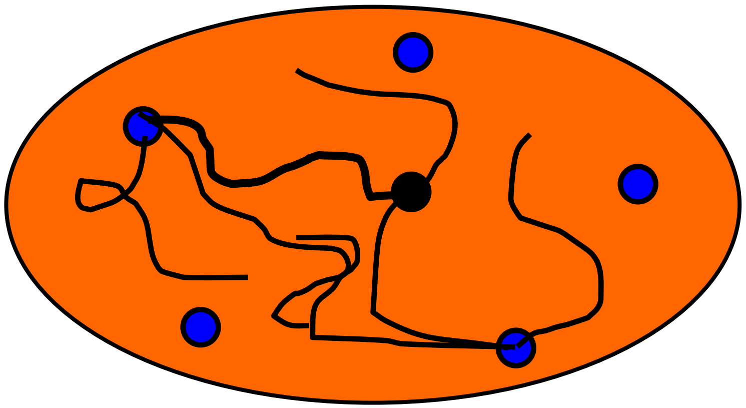
A decomposition statement



- [Spielman-Teng 04] Lovasz-Simonovitz curve technique for local partitioning
- [Kale-S-Peres 08] Understanding random walks with respect to behavior in subgraphs
  – Sublinear expander reconstruction (local algorithms to the rescue!)

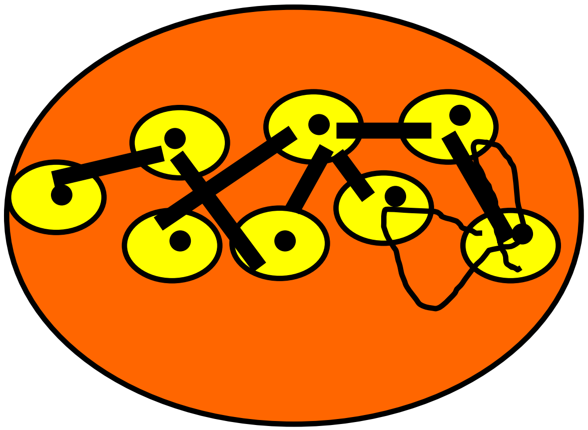
The algorithm (at long last)

If > n/ℓ leaky vertices



- Pick random s
- Perform O(1) poly(ℓ)-rws from s to get v_1, v_2…
- Perform $n^{1/2}$ poly(ℓ)-rws from each v_i, to get K_r minor

If < n/ℓ leaky vertices

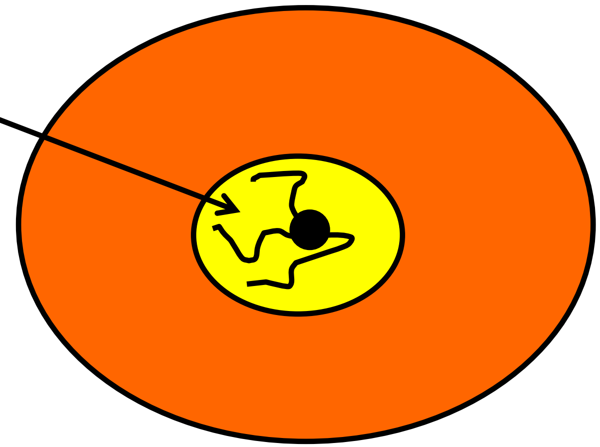


- Pick random s
- Perform poly(ℓ) ℓ-rws from s, and let S be set of vertices seen
- Use exact procedure to find H-minor in S

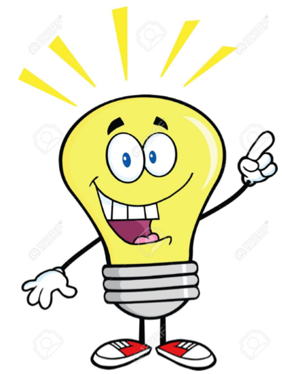
What about two-sided testers?

One-sided $\Longrightarrow$ Two-sided

At least ℓ^{10}



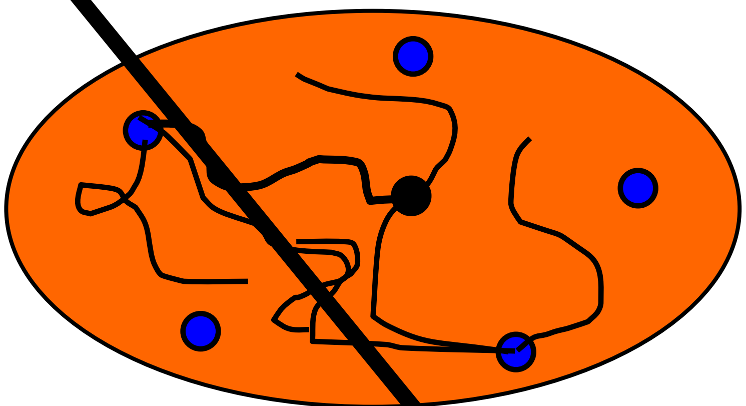
$$\|\mathbf{p}_{s,\ell}\|_2^2 \leq \ell^{-10}$$



- If there are at least n/ℓ leaky vertices, the random walk algorithm finds K_5 minor whp
- Cor: A planar graph has at most n/ℓ leaky vertices
  - Only need poly(ℓ) rws to test if vertex is leaky!

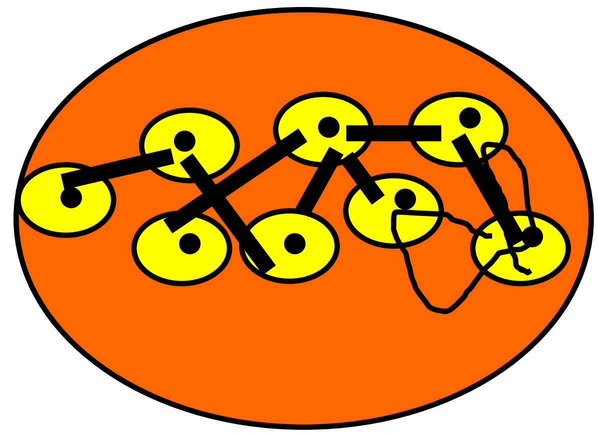
The two-sided tester

If > n/ℓ leaky vertices



- Pick random s
- Perform O(1) poly(ℓ)-rws from s to get v_1, v_2…
- Perform $n^{1/2}$ poly(ℓ)-rws from each v_i, to get K_r minor

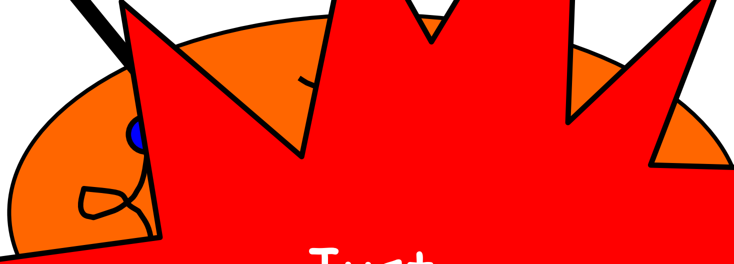
If < n/ℓ leaky vertices



- Pick random s
- Perform poly(ℓ) ℓ-rws from s, and let S be set of vertices seen
- Use exact procedure to find H-minor in S

The two-sided tester

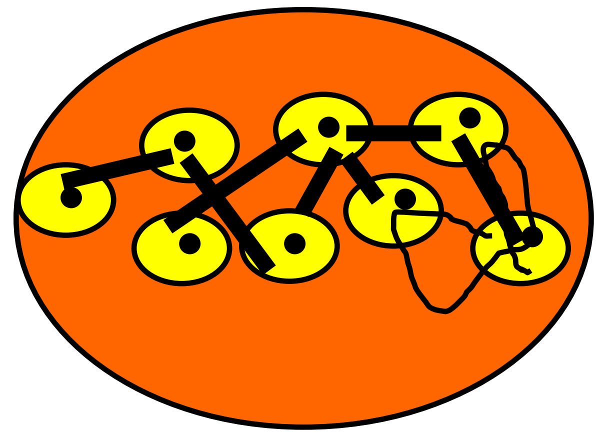
If > n/ℓ leaky vert…es



Just estimate number of leaky vertices

- … ly(ℓ)-rws
  fro… v_i, … get K_r minor

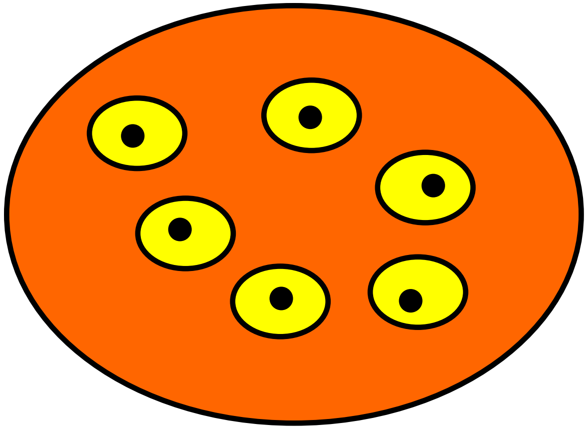
If < n/ℓ leaky vertices



- Pick random s
- Perform poly(ℓ) ℓ-rws from s, and let S be set of vertices seen
- Use exact procedure to find H-minor in S

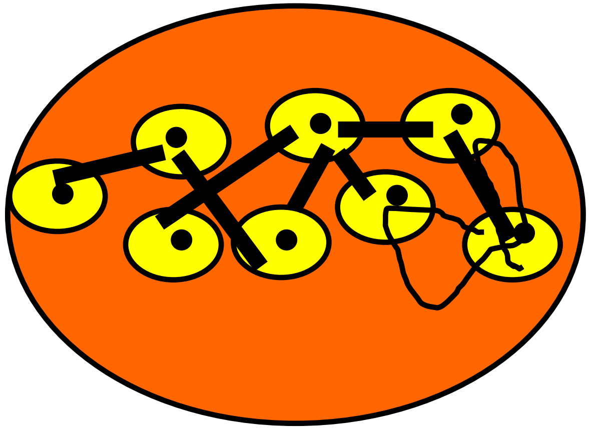
The two-sided tester

Estimate fraction of leaky vertices



- Pick poly(ℓ) random vertices s
- Perform poly(ℓ)-rws from each s to check if leaky
- Reject if $1/\ell$ fraction are leaky

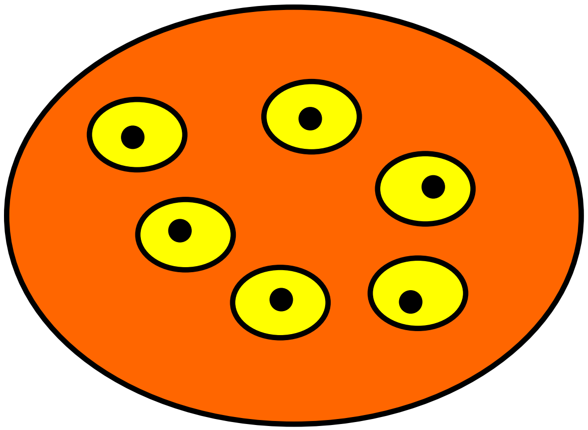
If pass, < n/ℓ leaky vertices $\Rightarrow$ decomposition exists



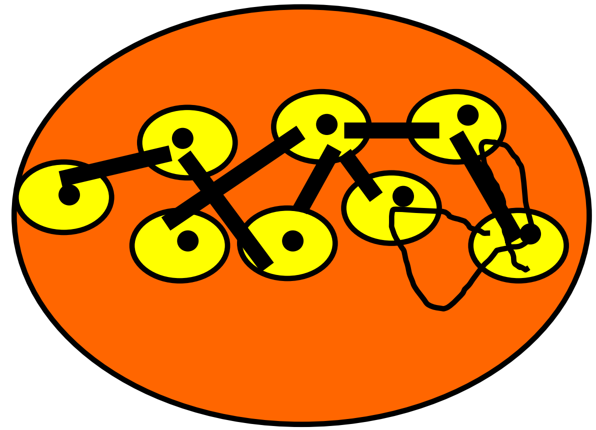
- Use exact procedure to find H-minor in subgraph visited

The two-sided tester

Estimate fraction of leaky vertices



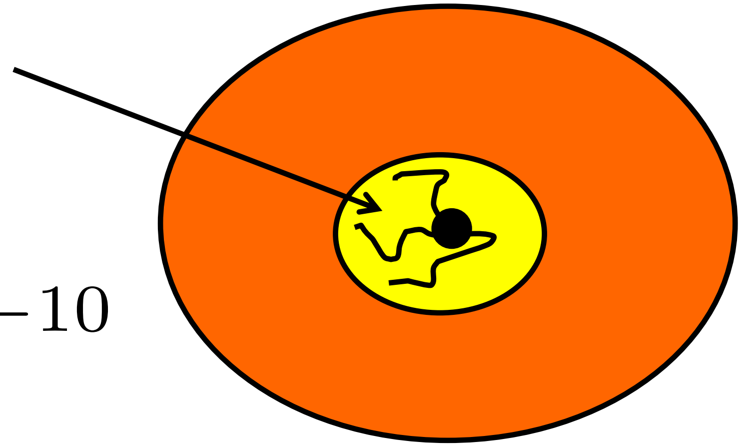
If pass, < n/ℓ leaky vertices
$\Rightarrow$ decomposition exists



- So you get poly(ℓ) tester
  - And $\ell = n^{\delta}$
- Argh! I need to set ℓ = poly(1/ε)

The length issue

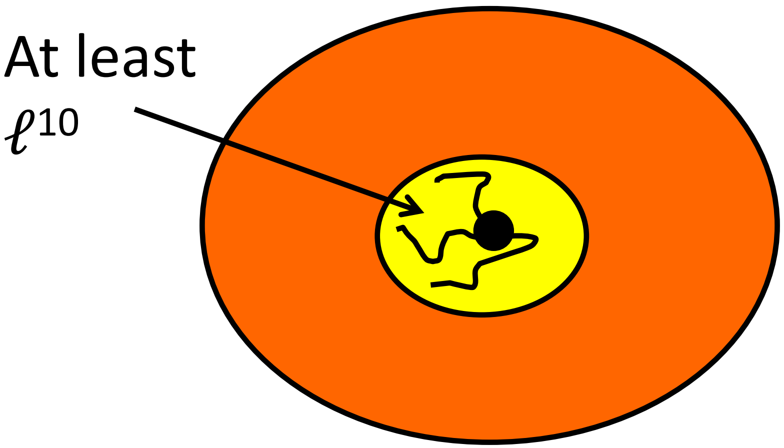
At least ℓ^{10}

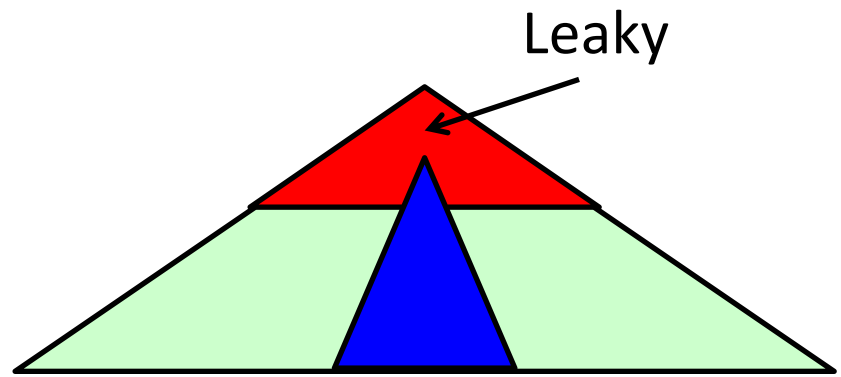


$$\|\mathbf{p}_{s,\ell}\|_2^2 \leq \ell^{-10}$$

- If there are at least n/ℓ leaky vertices, the random walk algorithm finds K_5 minor whp
- Cor: A planar graph has at most n/ℓ leaky vertices
- Proof needs ℓ > poly(log n)
  – Random walks have to be long enough

A direct proof

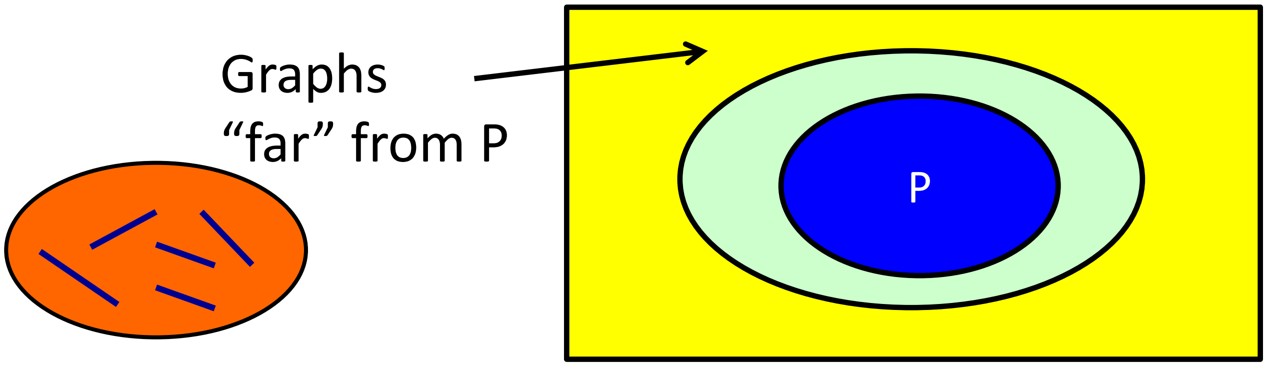


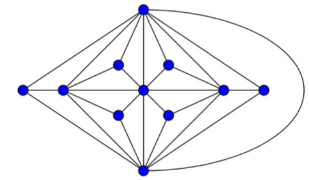


- Just prove the corollary directly
- Direct, shorter proof, with constant ℓ
  - Works for any hyperfinite property

- Thm: A planar graph has at most n/ℓ leaky vertices

And so…





Two-sided

- [Kumar-S-Stolman 19]

poly(1/ε) for all minor-closed properties

One-sided

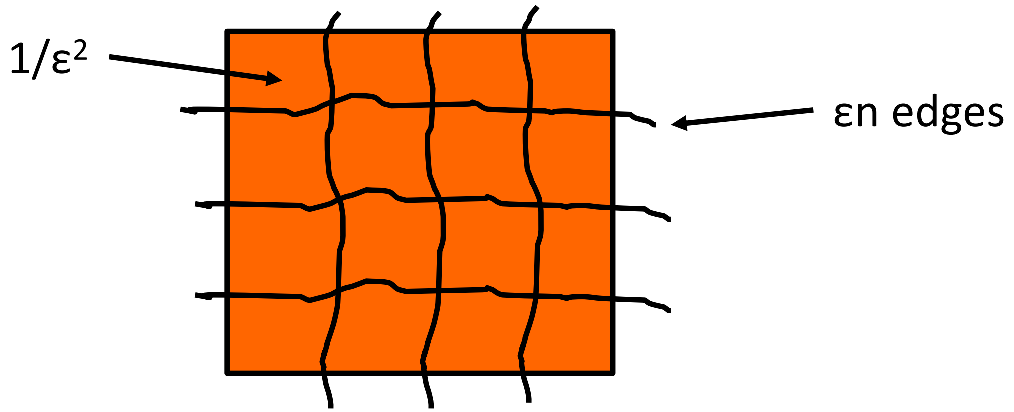
- [Kumar-S-Stolman 18]

$\sqrt{n} \cdot n^{o(1)}$ for all minor-closed properties

Based on (new?) toolkit using spectral graph theory for minor-freeness

What next?

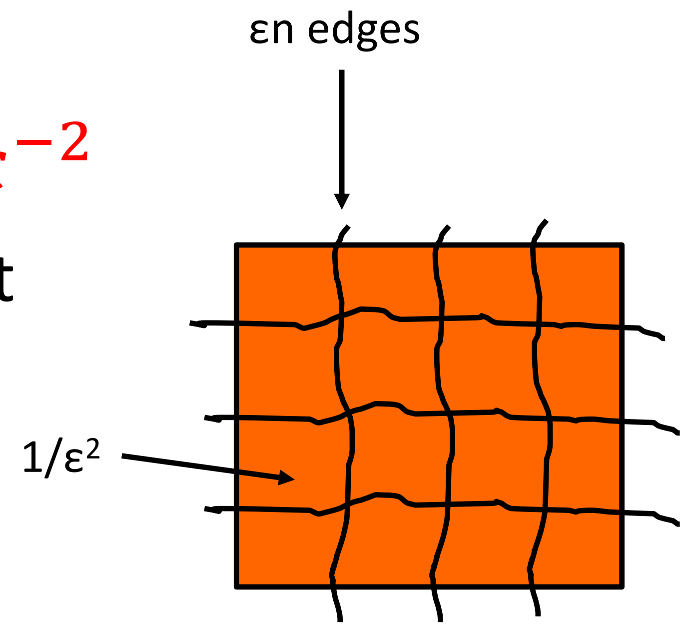
Partition oracles



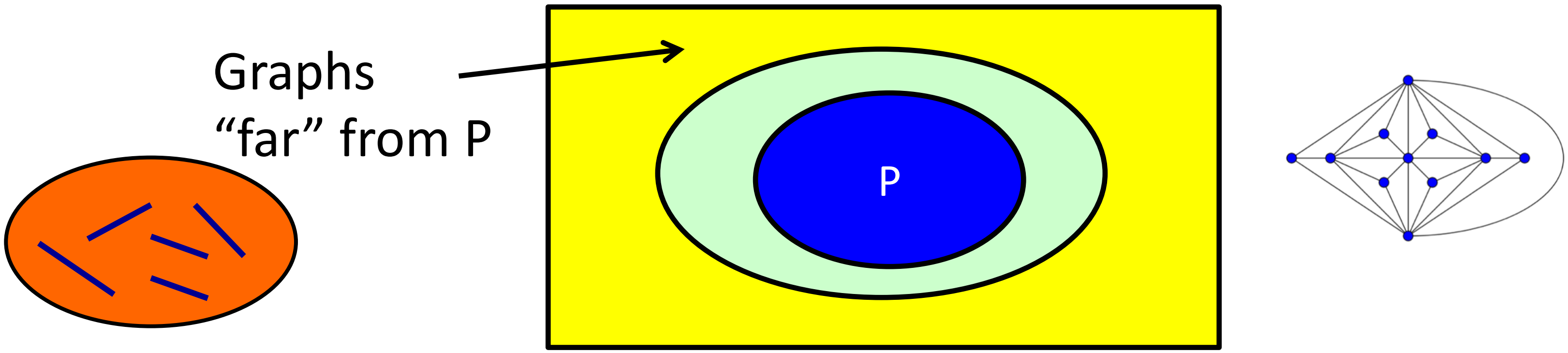
- Planarity is hyperfinite: remove εn edges to get connected components of poly(1/ε) size
- [Hassidim-Kelner-Nguyen-Onak 09, Levi-Ron 15] Query access to such a partition with no preprocessing!
  - But pieces/runtime of $(d/\varepsilon)^{\log(1/\varepsilon)}$ size
- Can we get partition oracle with runtime poly(d/ε)?

The right complexity?

- Currently: there is $d\varepsilon^{-100}$ time two-sided tester for P
- I think the right answer is $d\varepsilon^{-2}$
  - Not enough to tighten current proof



The degree dependence



Two-sided

- [Kumar-S-Stolman 19]

poly(d/ε) for all minor-closed properties

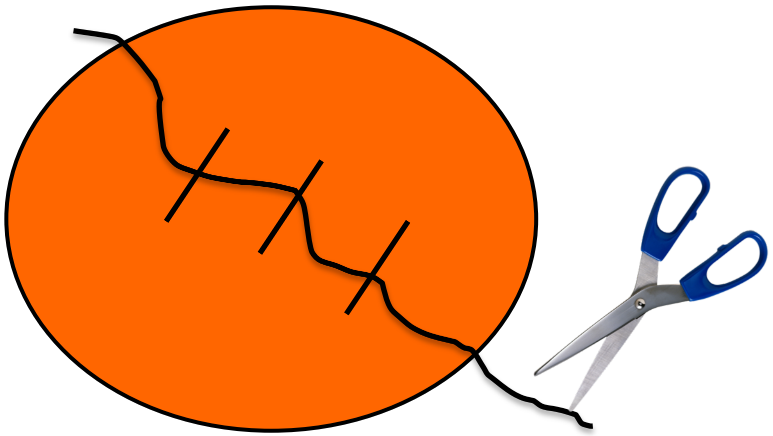
One-sided

- [Kumar-S-Stolman 18]

$d\sqrt{n} \cdot n^{o(1)}$ for all minor-closed properties

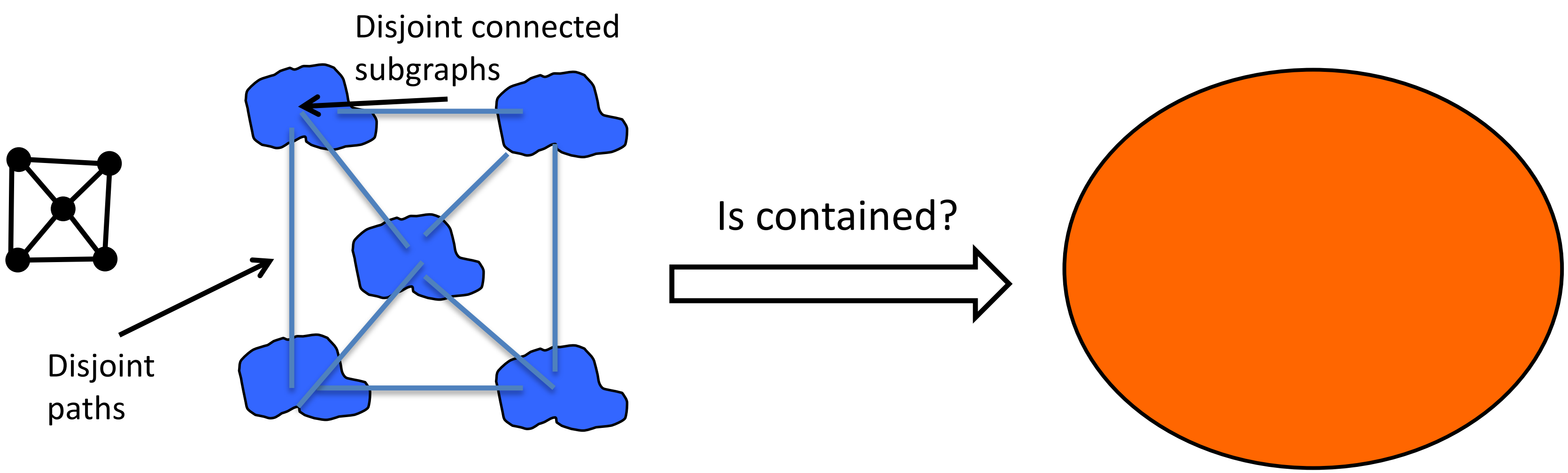
Can we make d the average degree, not the maximum degree?

Wishful thinking #1



- O(n) algorithms for $n^{1/2}$-sized balanced separators in H-minor free graphs?
- [Lipton-Tarjan79] O(n) for planar graphs
- [Alon-Seymour-Thomas 90] $O(n^2)$ algorithm
- [Plotkin-Rao-Smith 94] $O(n^{3/2})$ algorithm
- [Wulff-Nilsen 11] $O(hn^{5/4})$ algorithm
- [Kawarabayashi-Reed 10] $n^{1+\varepsilon}$ algorithm but tower dependence on |H|

Wishful thinking #2



- Deciding if G contains an H-minor
- [Kawarabayashi-Kobayashi-Reed12] $O(n^2)$ algorithm
- $o(n^2)$ algorithm using random walks?
- If graph has few leaky vertices, is the problem easier?

Thank you!