On the Densest *k*-Subgraph problem

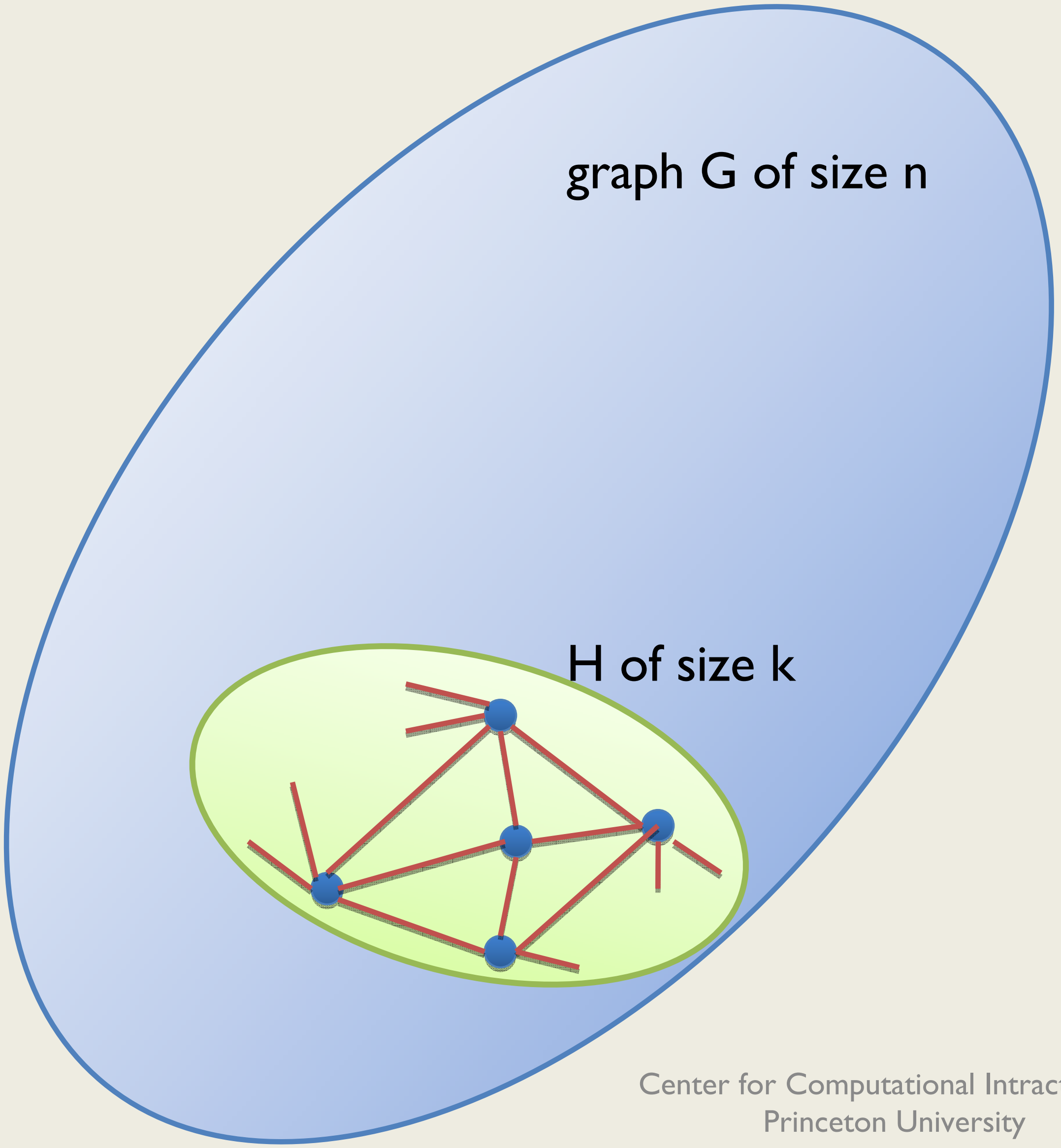
Aravindan Vijayaraghavan
Princeton University &
Center for Computational Intractability

Based on joint work

[Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uri Feige, V '10]

[Aditya Bhaskara, Moses Charikar, Venkat Guruswami, V, Yuan Zhou '11]

The Dense Subgraph Problem



Problem. Given *G*, find a subgraph H of size *k* of max. density (think of *k* as n^{ρ}).

Notation:

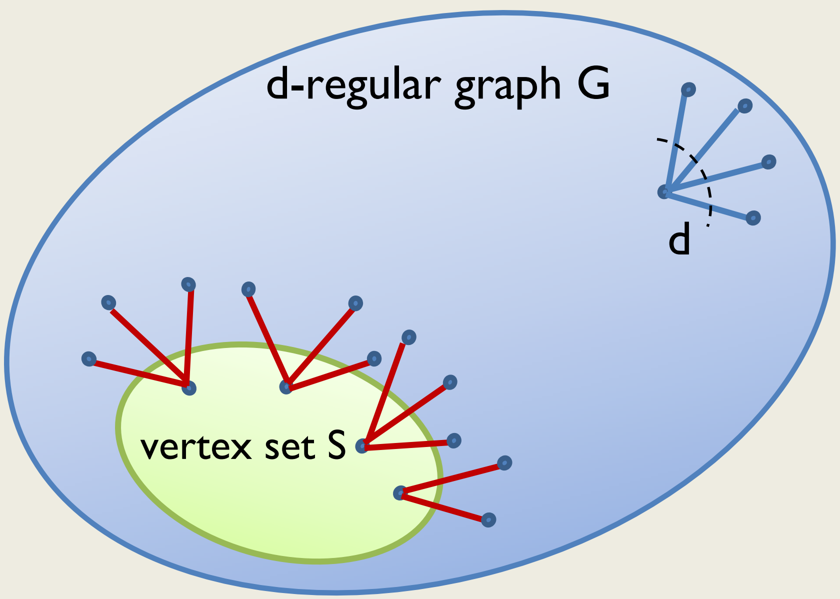
Density (H) = Avg. degree in induced subgraph H

Related problems

- Max–density subgraph (no size restriction):

  Polynomial time algorithm [GGT'87]

- Small set expansion



$$\text{Expansion}(S) = \frac{\#\ \text{edges leaving } S}{d\,|S|}$$

Dense subgraphs are everywhere !

A useful subroutine for many applications

- Social networks: Trawling the Web for emerging cyber-communities [KRRT '99]
  - *Web communities are characterized by dense bipartite subgraphs*
- Computational biology: Mining dense subgraphs across massive biological networks for functional discovery [HYHHZ '05]
  - *dense protein interaction subgraph corresponds to a protein complex* [BD'03] [SM'03][SS '05]

Dense subgraphs are everywhere !

- A useful subroutine for many applications.

- A useful candidate hard problem with many consequences

Average case hardness assumption

- [ABW '10] Variant was used as the hardness assumption in Public Key Cryptography.

  Non-expanding small set – private key.
- [ABBG'10] Toxic assets can be hidden in complex financial derivatives to commit undetectable fraud
- [KZ'11,CMVZ'11] Evidence of inapproximability for many problems assuming hardness of planted variants.

How does DkS fit in?

Densest k-subgraph
as a CSP with a strict
budget:

DkS = (trivial) Max 2-AND
at most k-variables
set to 1

Reeling in the years…

Problem. Given *G*, find a subgraph of size *k* with the maximum number of edges (think of *k* as n^{ρ})

Algorithms:

[FKP 93] give an $O(n^{1/3 - 1/90})$ approximation algorithm

Inapproximability:

[Feige 03] No PTAS under the Random 3-SAT assumption

[Khot 05] No PTAS unless NP $\subseteq$ BPTIME(sub-exp)

[RS 10] No constant factor approx assuming Small Set Expansion Conjecture

[FS 97] Natural SDP has an $\Omega(n^{1/3})$ integrality gap

Algorithm

[Bhaskara, Charikar, Chlamtac, Feige, V'10]

Theorem. $O(n^{1/4+\varepsilon})$ approximation for DkS in time $O(n^{1/\varepsilon})$

(Informal) Theorem. Can efficiently detect subgraphs of high log-density.

Strong Hierarchy Integrality gaps

[Bhaskara, Charikar, Guruswami,V, Zhou'11]

Theorem. $\tilde{\Omega}(n^{1/4})$ approximation for DkS for $\Omega(\log n/\log\log n)$ levels of SA+ (Sherali-Adams +SDP)

Theorem. $n^{\Omega(\varepsilon)}$ gap for $n^{1-\varepsilon}$ levels of Lasserre hierarchy

Outline

- Notion of log-density
- Algorithms for DkS:
  § Planted DkS: 'Local counting' based algorithms.
  § LP hierarchies to imitate arguments in worst case.
- Integrality gaps for strong hierarchies
- Open problems

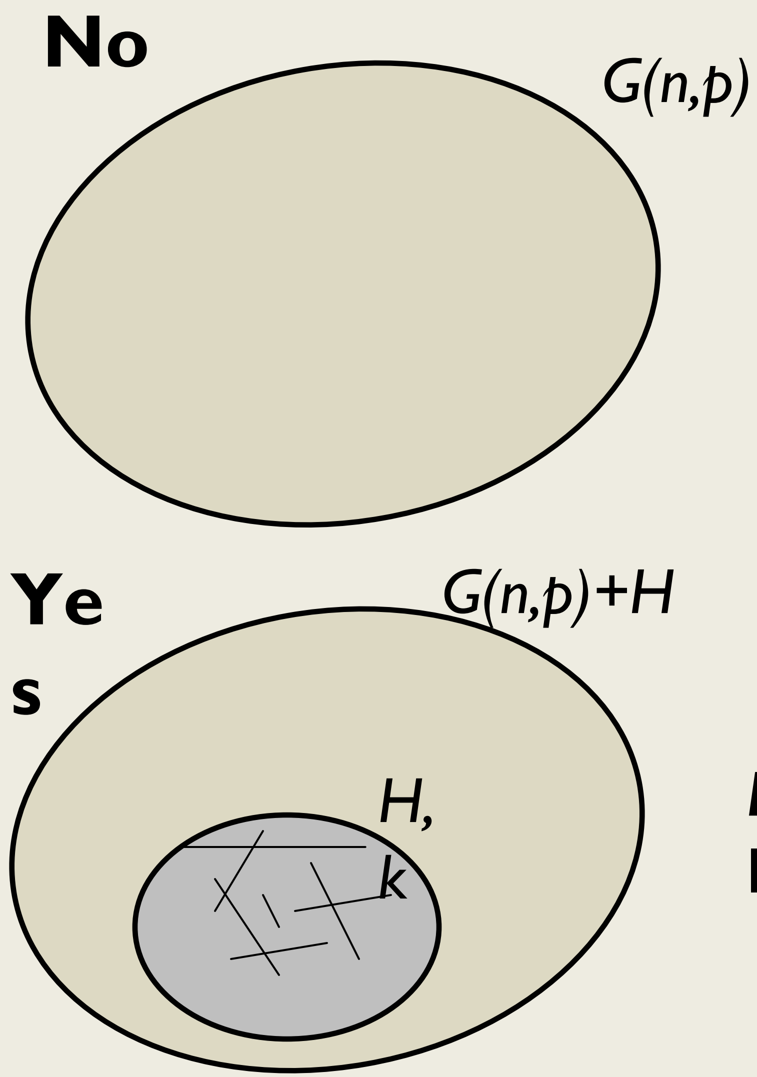
Log density

A graph on n vertices has **log-density** δ if the average degree is n^{δ}

$$\delta = \frac{\log d_{avg}}{\log |V|}$$

Question. Given G, can we detect the presence of a subgraph on k vertices, with higher log-density?

Planted versions of DkS



- Assume *G* does not have dense subgraphs
- Good algorithm for DkS $\Rightarrow$ we can distinguish

Problem. Distinguish between

NO: *G(n,p) of log-density* δ

YES: *G(n,p)* (same p) with *k*-subgraph of *log-density* $\delta+\epsilon$

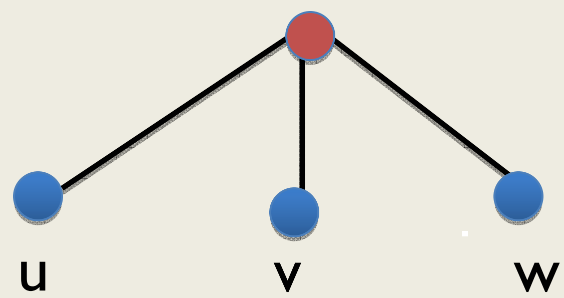
Note:

In *G(n,p)*, a k-subgraph H has density~ *kp*

$= k\ (n^{\delta}/n) < k^{\delta}$

Main idea

Example. Say δ = 2/3, i.e., degree = $n^{2/3}$ ($p=n^{-1/3}$)



random graph $G(n, n^{-1/3})$:

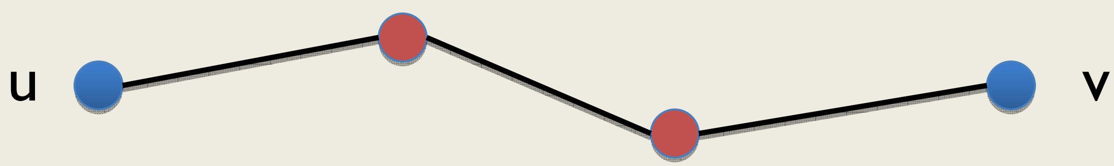
any three vertices have O(log n) common neighbors w.h.p. ($n.p^3$ in expectation)

planted graph: size k, log-density $2/3+\varepsilon$:

exists triple with $k^{3\varepsilon}$ common neighbors

Main idea (contd.)

Example 2. δ = 1/3, i.e., degree = $n^{1/3}$ ($p = n^{-2/3}$)



random graph $G(n, n^{-1/3})$:

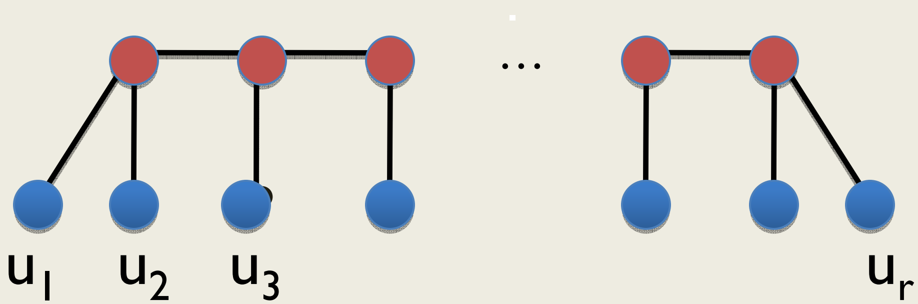
any pair of vertices have O($\log^2 n$) **paths of length 3**, w.h.p. (n^2p^3 in expectation)

planted graph: size k, log-density 1/3+ε:

exists a pair of vertices with k^ε paths

Main idea (contd.)

General strategy: For each rational δ, consider appropriate `caterpillar' structures, count how many `supported' on fixed set of leaves



§ Random graph *G(n,p)*, log-density δ:

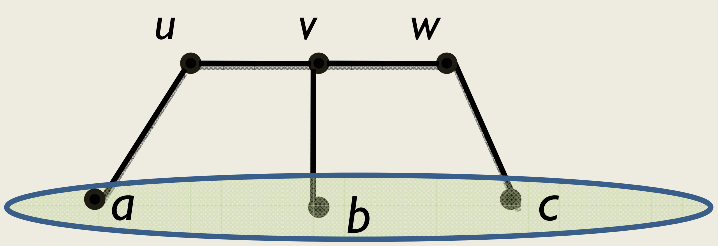
every leaf tuple supports polylog(*n*) caterpillars

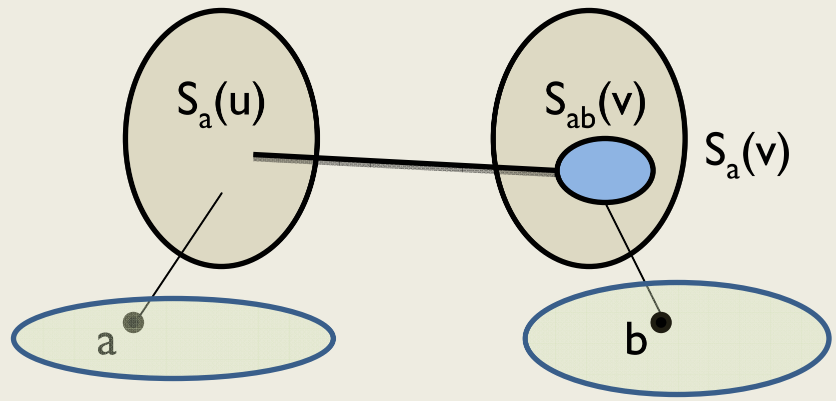
§ Planted graph, size *k*, log-density δ+ε :

some leaf tuple supports at least k^{ε} caterpillars

Analysis for NO case (δ = 2/5 i.e. $p=n^{-3/5}$)

TO SHOW: *Every* leaf tuple supports polylog(*n*) caterpillars





Idea: Upper bound #candidates for each internal node by polylog(n).

Fix tuple (a,b,c). Eg: $S_{ab}(v)$ -- candidates for v after fixing a,b.

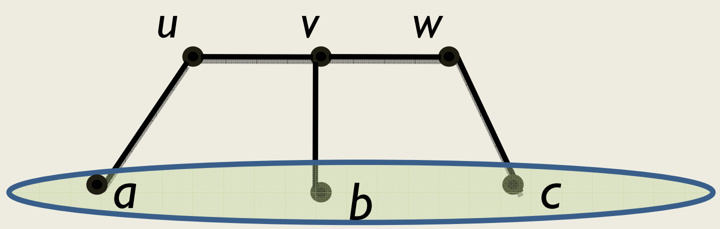
$\mathbf{E}[|S_a(u)|] \sim D = np = n^{2/5}$, and it is concentrated.

Similarly, $\mathbf{E}[|S_a(v)|] \sim n^{4/5}$ and concentrated.

$\mathbf{E}[|S_{ab}(v)|] \sim n^{4/5}p \sim n^{1/5}$ and it's concentrated.

Similarly, $\mathbf{E}[|S_{abc}(w)|] \sim n^{1/5}.\ np.\ p = O(1)$

Proof for $\delta = 2/5$



- # of "candidate *w*'s" given leaves *a,b,c* is < log *n* w.h.p.
- The same is true for "candidate *v*'s and u's" too by similar arguments.

Thus the number of structures is < ($\log^4 n$) w.h.p.

Dense vs. Random – conclusion

Theorem. For every $\epsilon > 0$, and $0<\delta<1$, we can distinguish between *G(n,p)* of log-density δ, and a graph with a *k*-subgraph of log-density $\delta+\epsilon$, in time $n^{O(1/\epsilon)}$.

(Pick a rational no. in $[\delta,\delta+\epsilon)$ and use the appropriate caterpillar)

- k-subgraphs in G(n,p) have density $\max\{1, kn^{\delta}/n\}$
- Can detect planted k-subgraphs of density $k^{\delta+}$

- *Distinguishing ratio* ~ $\max_{\delta,k} \dfrac{k^{\delta}}{\max\{1, kn^{\delta}/n\}} = O(n^{1/4})$

DkS in general graphs

Moving from average case to worst case

DkS in general graphs

Input. G on n vertices, degree $\leq D$

Promise. There is a subgraph H on k vertices with average degree d

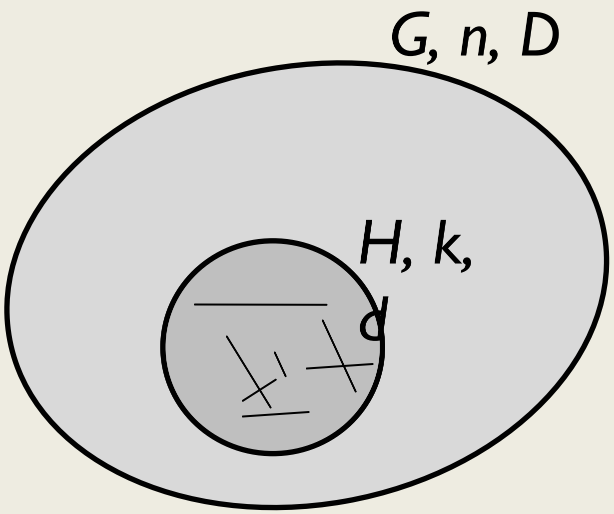
Question. How dense a k-subgraph can we find?

An algorithm in worst case by mimicking our distinguishing algorithm for random graphs.

Some simplifications

Given: *A* regular graph G with degree D= n^{δ} such that k.D=n

(k-subgraph in G has ~ O(1) density.)

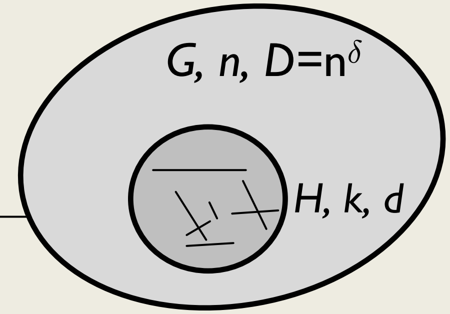


H is k-subgraph of G with min-degree $d=k^{\delta+\epsilon}$ (higher log-density)

Aim: Enough to output a k-subgraph of density ρ (ρ is a large constant)

Observation: Can return a ρ-dense subgraph with $\leq$ k vertices (remove, repeat)

An outline of the algorithm



- Inspired by algorithm for Planted problem.
- Algorithm for each δ uses the structure Cat_{δ} (size s_{δ})

Algorithm proceeds for s_{δ} steps.

Idea. Look at the 'set of candidates' for a non-leaf after *fixing* a prefix of the leaves

S_t -- candidate vertices at step t of the caterpillar.

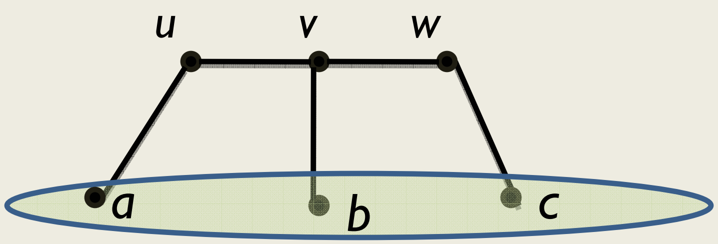
LP(S) -- the number of vertices from H in S.

Algorithm either finds dense-subgraph from S_t or

It *'behaves'* as in random case and lower bound $LP(S_{t+1})/|S_{t+1}|$

Finally, $LP(S_t)/|S_t|>1$ (contradiction) ……

Algorithm using Cat_δ (plot outline)



Procedure LocalSearch(S)
Tries to find a dense subgraph greedily between **S** and Γ(**S**)

1. S_0=V. Perform LocalSearch(S_0)
2. If we don't get a dense subgraph,
   then $\exists$ ***a*** s.t. $|S_a(u)| \leq U_1$ (as in random graph) and $|LP(S_a(u))| \geq L_1$.
3. Do LocalSearch($S_a(u)$). If it fails then $|S_a(v)| \leq U_2$ and $|LP(S_a(v))| \geq L_2$
4. Do LocalSearch($S_a(v)$). If fail, $\exists$ ***b*** s.t bounds like random

Keep doing this … At the last step, the parameters give a contradiction!

LP relaxation (a hierarchy) for Cat_δ

Intended solution: k-subgraph H with minimum degree d

Simple LP:

$$\sum_{i \in V} y_i \leq k \quad \text{and} \qquad (1)$$ (size at most k)

$$\exists y_{ij} \; : \; i,j \in V \quad \text{s.t.}$$

$$\forall i \in V \quad \sum_{j \in \Gamma(i)} y_{ij} \geq d y_i \qquad (2)$$ (min degree d in H)

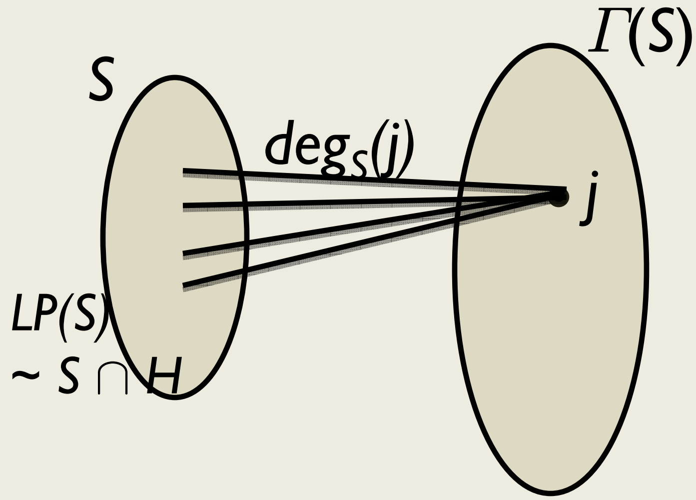
$$\forall i,j \in V \quad y_{ij} = y_{ji} \qquad (3)$$

$$\forall i,j \in V \quad 0 \leq y_{ij} \leq y_i \leq 1 \qquad (4)$$

LP : Simple LP + LS hierarchy for s_δ levels.

- Captures fixing leaves since $\{y_{ij} / y_j\}$ satisfy LP too.
- LP is feasible for any constant number of conditionings (i.e. fixing leaves).

Main Component – LocalSearch(S)



Consider $k' = LP(\Gamma(S))$ (<= k)

$$\text{Edges}(S, S_{k'}) \geq \sum_{j \in \Gamma(S)} y_j \deg_S(j)$$

$$\geq \sum_{i \in S} \sum_{j \in \Gamma(i)} y_{ij} \geq d\,LP(S)$$

(due to eq 2)

<u>Greedy algorithm:</u>

For each $k' = 1 \ldots k$, do:

- $S_{k'}$ = k' vertices in $\Gamma(S)$ with the most edges to S.
- Let S* be k vertices from S with most edges into $S_{k'}$.

If $S_{k'}$ U S* has density $\geq \rho$, return it.

If no ρ dense subgraph is found, return Fail

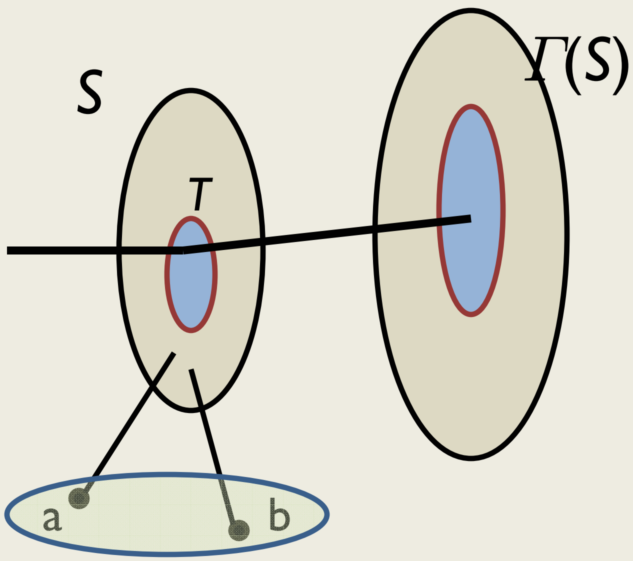
Lem. LocalSearch finds a graph of density at least $= \dfrac{d\,LP(S)}{LP(\Gamma(S)) + |S|}$

Round or Bound -1 (backbone edge)

Claim1: Let S be candidates, $\{y_i\}$ be LP solution, we either

a) Output a k-subgraph of density ρ using LocalSearch

b) else $LP(\Gamma(S)) \geq d\ LP(S)/\rho$ (we can set $S_{new} = \Gamma(S)$)



If we do not find ρ dense subgraph,

$S_{new} = \Gamma(S)$

LP($\Gamma(S)$) increases by at least d/ρ

and $|\Gamma(S)|$ increases by at most D

(like in the random case)

Round or Bound – 2 (leaf/hair)

Claim 2: If S is candidate set, $\{y_i\}$ is LP solution, we either

a) Find a k-subgraph of density ρ between S and $\Gamma(S)$

b) or find leaf j if $S_{new}=S \cap \Gamma(j)$

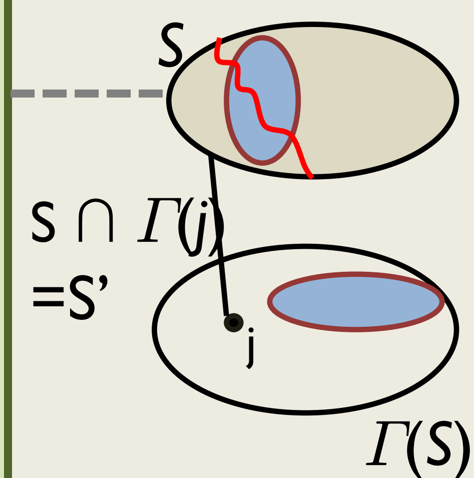
$LP(S_{new}) \geq d\, LP(S)/2k$ *and* $|S_{new}| \leq \rho(|S|+k)/k$

If we do not find a dense subgraph,

$$\rho(|S|+k) \geq \sum_{j \in \Gamma(S)} y_j |S \cap \Gamma(j)| \geq \sum y_j \, LP_{\{y_{ij}/y_j\}}(S \cap \Gamma(j))$$

$$= \sum_{j \in \Gamma(S)} \sum_{i \in S \cap \Gamma(j)} y_{ij} \geq d LP(S)$$

By averaging argument, we can pick $j \in \Gamma(S)$ such that Claim follows.



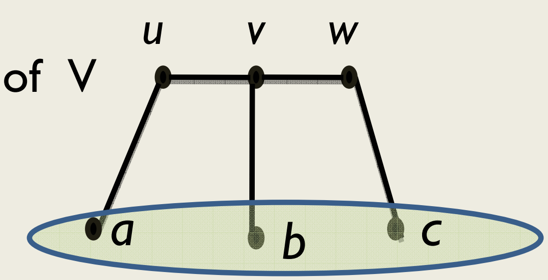
To summarize…

Roughly speaking, if we don't find a dense subgraph in a step,

- every backbone step, LP(S)/|S| decreases by O(d/D)
- every hair step, LP(S)/|S| increases by at least Ω(d)

Because of choice of structure, LP(S)/|S| becomes >1 at final step (a contradiction).

Completing the algorithm for $\delta = 2/5$



1. $S_0 = V$. $LP(S_0)/|S_0| = k/n$.
2. If LocalSearch(S_0) doesn't give
   a 100-dense subgraph,
   ∃ ***a*** to condition on so that,
   $LP(S_a(u))/|S_a(u)| \geq dk/n$
3. If LocalSearch($S_a(u)$) fails, $LP(S_a(v))/|S_a(v)| \geq d^2k/Dn$
4. If LocalSearch($S_a(v)$) fails, ∃ ***b*** $LP(S_{ab}(v))/|S_{ab}(v)| \geq d^3k/Dn$.
5. If LocalSearch($S_{ab}(v)$) fails, $LP(S_{ab}(w))/|S_{ab}(w)| \geq d^4k/D^2n$
6. If LocalSearch($S_{ab}(w)$) fails, $LP(S_{abc}(w))/|S_{abc}(w)| \geq d^5k^3/n^3 > 1$

(a contradiction)

Beating the log-density barrier?

- $n^{(1-\varepsilon)/4}$ approximation in time $2^{n^{6\varepsilon}}$
- Guess subsets of size n^{ε} for every leaf in caterpillar structure.
- Integrality gaps suggest polytime algorithms from Sherali-Adams (SA+) relaxations can not beat the barrier.

Stronger relaxations

Lasserre

Sherali-Adams

Lovasz-Schrijver

Gaps for lift-and-project

[BCCFV '10]

t rounds of Lovasz-Schrijver: gap $n^{\frac{1}{4}+O(1/t)}$

[BCGVZ '11]

- $\Omega(\frac{\log n}{\log \log n})$ levels of Sherali-Adams:
  gap $\tilde{\Omega}(n^{\frac{1}{4}})$
- $n^{\Omega(1)}$ levels of Lasserre: $n^{\Omega(1)}$ gap

Lasserre gaps

- First constructs gaps for Max r-CSP(q) instances over large alphabet size r,q= $n^{\Omega(1)}$.
- Simple reduction from Max r-CSP(q) to DkS
- Uses Tulsiani's framework to transform the Lasserre gaps for DkS.

Small Set Expansion (SSE) problem [RS '10]

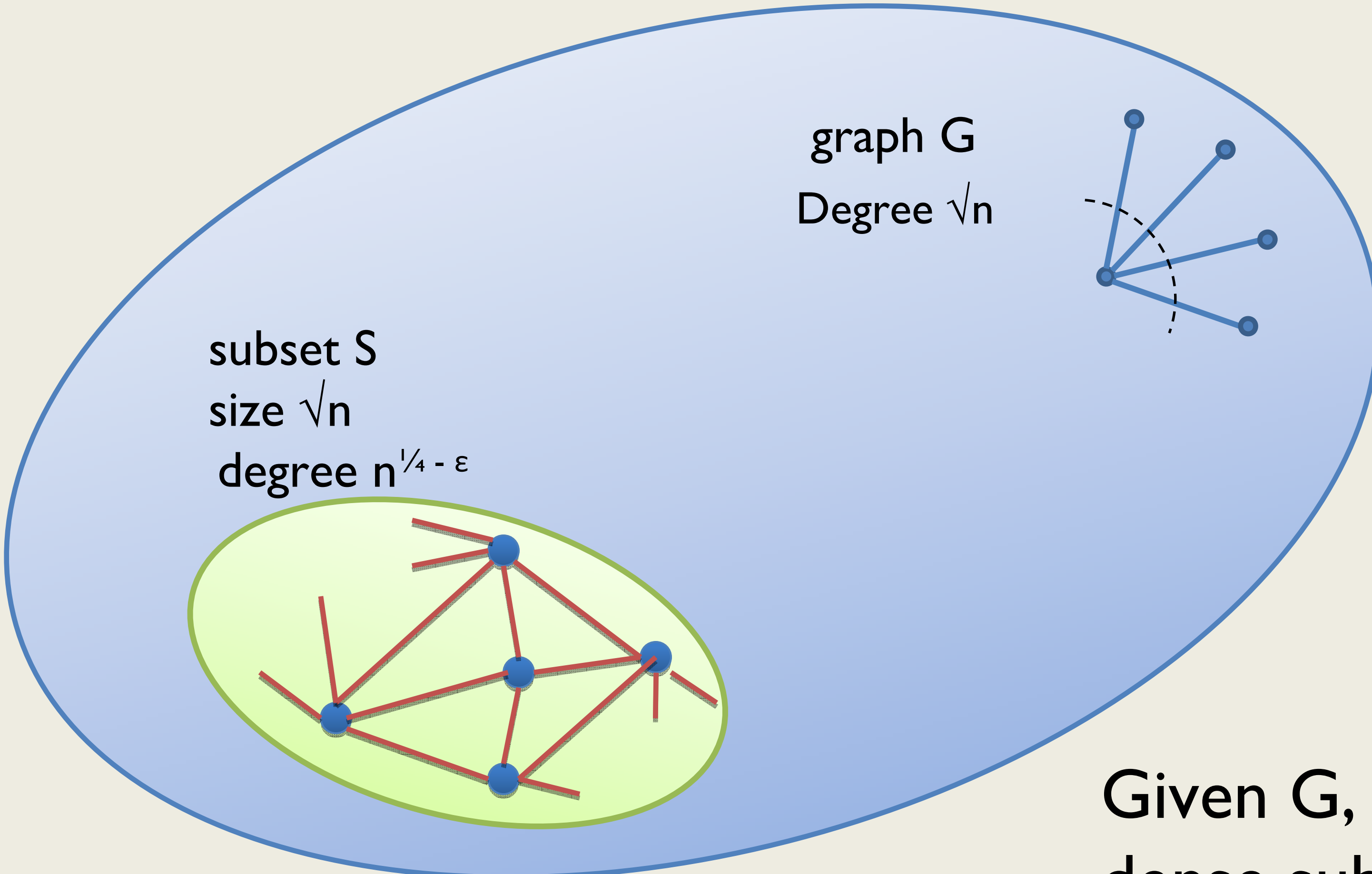
Given ε,δ> 0, D-regular graph G, distinguish between
(think of D as constant)

- [NO] Small subgraphs expand very well
  Every subgraph of size ≤δn has density ≤ εD
- [YES] Some small subgraph does not expand
  Some subgraph of size ≤δn has density ≥ (1-ε)D

[ABS'10] $\exp(n^{O(poly(\varepsilon))})$ time algorithm for SSE.
[BRS'11,GS'11] distinguish using $n^{O(poly(\varepsilon))}$ levels of Lasserre.

$n^{\Omega(1)}$ levels Lasserre gap for DkS seems to suggest that DkS much harder than SSE

Open Problem



Given G, find dense subgraph S

Open Problems

- Better algorithms using SDPs in certain ranges of parameters? (like [Steurer'11])
- Evidence of large inapproximability of DkS?
- Stronger integrality gaps?

  Maybe $n^{1/4-\varepsilon}$ gap for n^{ε} levels of the hierarchy?

Thank you!