Beyond Worst Case Analysis in Approximation

Uriel Feige

The Weizmann Institute

Plan of talk

Survey some known approximation algorithms and open questions for worst case and random instances of:

- max-3SAT
- min-bisection
- 3-coloring
- unique games
- dense k-subgraph

A question to keep in mind

Does the study of algorithms that handle random inputs help in designing approximation algorithms for worst case instances?

Max 3-SAT

A 3-CNF formula with n variables and m clauses

$(\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee x_4) \wedge \ldots$

Find an assignment that maximizes the number of clauses satisfied.

A random assignment satisfies $7/8\, m$ clauses in expectation.
Gives approximation ratio $7/8$.
Achieving an approximation ratio of $\rho > 7/8$ is NP-hard [Hastad 1997, 2001].

Random max-3SAT

Each literal in input 3CNF formula chosen uniformly at random.

Approximation algorithm with ratio ρ for random instances:

- If it outputs an assignment, then the number of clauses satisfied by the assignment is guaranteed to be at least $\rho *$ opt.
- Allowed to say "don't know" with probability at most $1/2$ (over choice of random input).

No algorithm is known (or even conjectured) to achieve an approximation ratio better than $7/8$ on random instances with $m \gg n$.

Random instances appear to be as difficult as worst case instances

Max 3-SAT is NP-hard to approximate with a ratio better than $7/8$.

There are distributions over random instances for which we do not know how to obtain an approximation ratio better than $7/8$.

Some questions:

Max 3-SAT is NP-hard to approximate with a ratio better than $7/8$.

There are distributions over random instances for which we do not know how to obtain an approximation ratio better than $7/8$.

Can we prove NP-hardness for random instances?

Some questions:

Max 3-SAT is NP-hard to approximate with a ratio better than $7/8$.

There are distributions over random instances for which we do not know how to obtain an approximation ratio better than $7/8$.

Can we prove NP-hardness for random instances? Currently, no.

Some questions:

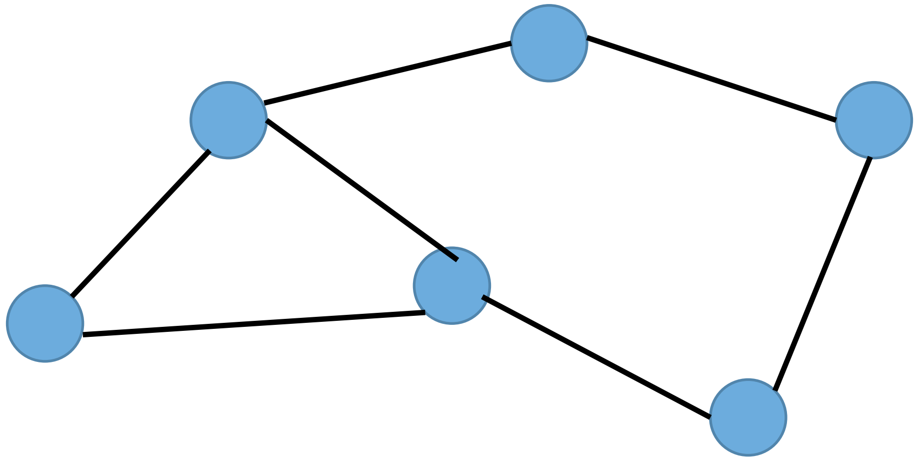
Max 3-SAT is NP-hard to approximate with a ratio better than $7/8$.

There are distributions over random instances for which we do not know how to obtain an approximation ratio better than $7/8$.

Suppose that a problem is NP-hard to approximate within a ratio better than ρ. Is there a natural (sampleable) distribution over inputs on which it is hard to achieve an approximation ratio better than ρ?

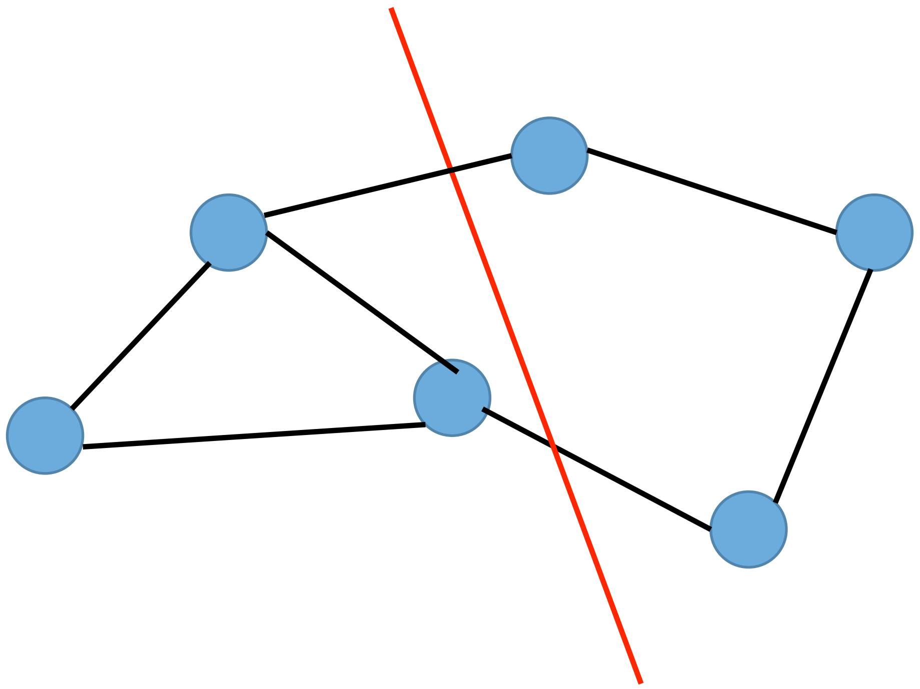
Min-bisection

Partition an n-vertex graph into two equal size parts, minimizing the number of edges in the cut.



Min-bisection

Partition an n-vertex graph into two equal size parts, minimizing the number of edges in the cut.



Known results

- Approximable within $O(\log n)$ [Racke 2008]
- For some $\rho>1$, ETH-hard to approximate [Khot 2004, 2006]

Bi-criteria approximation (allowed to output a nearly balanced cut):

- Within $O(\sqrt{\log n})$ [Arora, Rao, Vazirani 2004, 2009]
- For some $\rho>1$, ETH-hard to bi-approximate [Ambuhl, Mastorlili, Svensson 2007, 2011]

Random instances of bisection

Random graph with $m \gg n$ edges.

Minimum bisection is only slightly smaller than $m/2$.

Can indeed certify this in polynomial time using a spectral algorithm:

- Random graph is nearly d-regular for $d=2m/n$.
- Largest eigenvalue of adjacency matrix is roughly d.
- Second largest eigenvalue of adjacency matrix is $O(\sqrt{d}\,)$ (w.h.p.).
- Had there been a small bisection, there would have been at least two $\Omega(d)$ eigenvalues.

Approximation ratio nearly 1 on random instances.

Other distributions of random graphs

For almost all (sufficiently dense) graphs with a minimum bisection significantly smaller than $m/2$, can find the minimum bisection in polynomial time and certify its minimality [Boppana 1987]. Uses semidefinite programming (SDP), an algorithmic technique that extends both linear programming and spectral algorithms.

Is there a distribution over graphs for which it seems plausible that achieving a constant factor approximation is hard?

Algorithmic connections

The current best bi-criteria approximation [Arora, Rao, Vazirani] uses SDPs, which are used also for random instances.

The previous best (true) approximation [Feige, Krauthgamer 2000, 2002] uses the bi-criteria ones as a blackbox (at an $O(\log n)$ multiplicative loss in the approximation ratio).

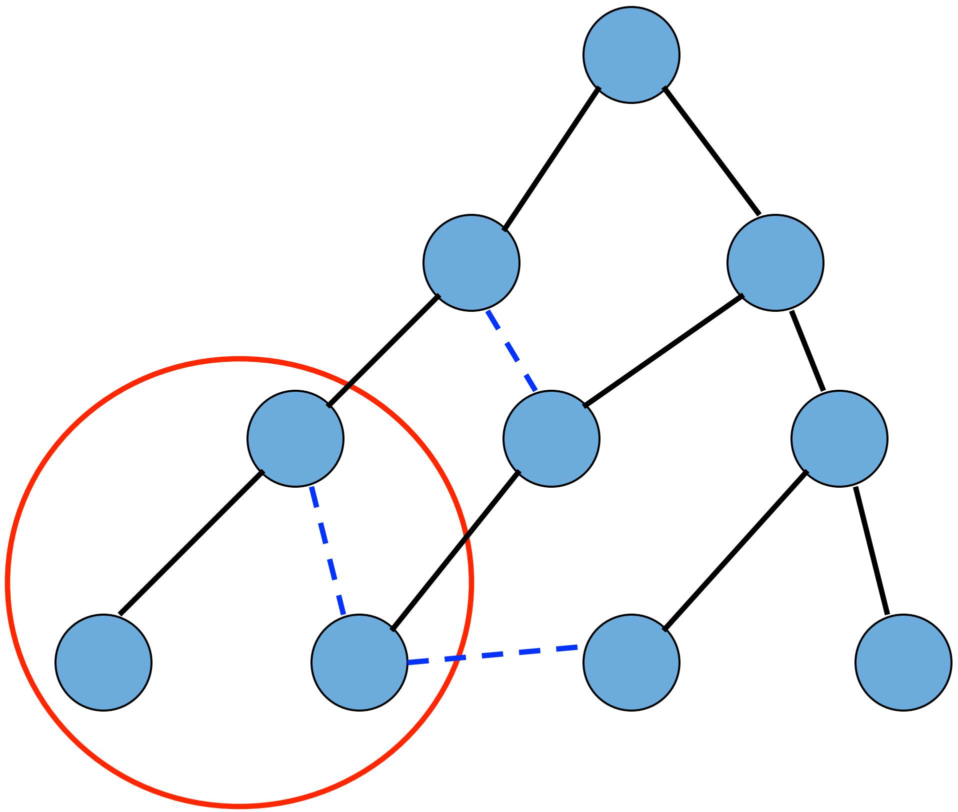
The current best (true) approximation [Racke 2008] does not use SDPs. It is based on randomized embeddings into trees, where every edge suffers an average load of $O(\log n)$.

The load on edges in a spanning tree

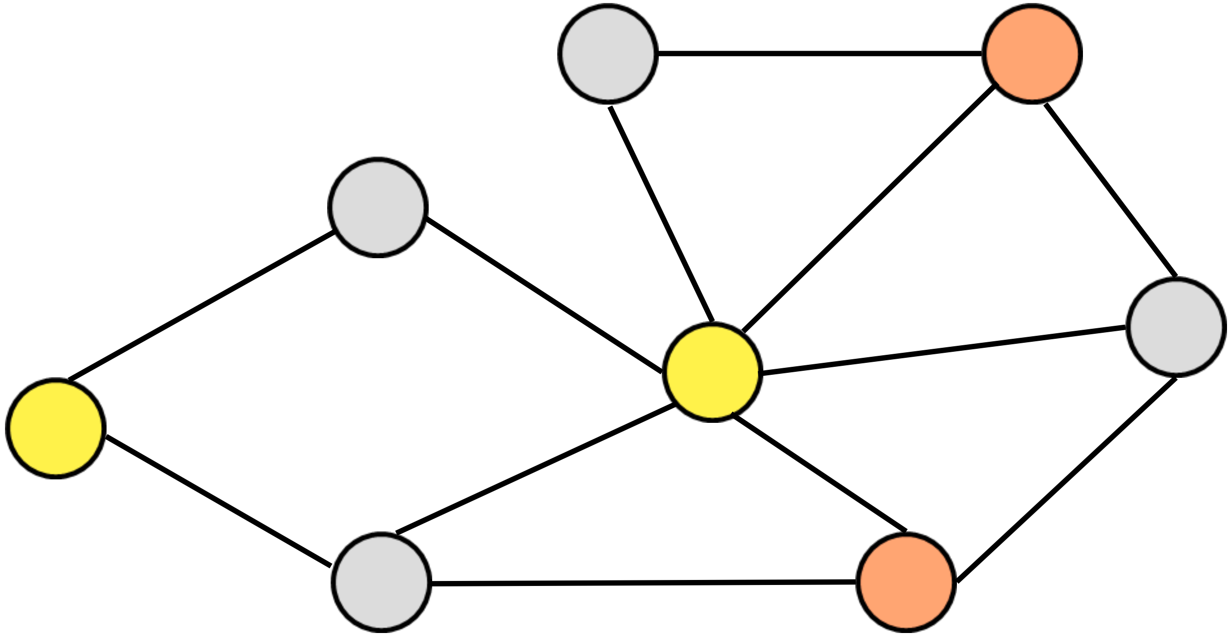
The cut contains:

- 2 spanning tree edges
- 3 graph edges

However, its load is 4.



3-coloring



Min 3-coloring

Given a 3-colorable graph, legally color it with few colors.

NP-hard to 4-color [Khanna, Linial, Safra 1993, 2000].

Graphs of maximum degree d (that may depend on n):

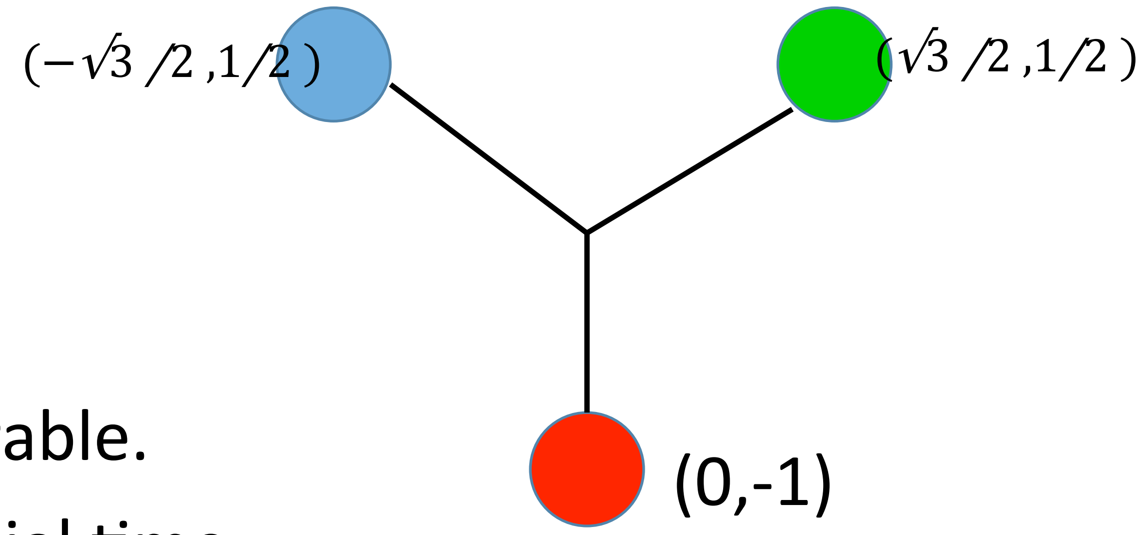
- Greedy coloring uses at most $d+1$ colors.
- [Karger, Motwani, Sudan 1994, 1998]: a polynomial time algorithm that colors graphs that satisfy the vector 3-coloring SDP relaxation, using $O^*(d^{1/3})$ colors.

Vector 3-coloring

v_i - unit vector for vertex i

$v_i v_j \leq -1/2$ if $(i,j) \in E$.

$v_i v_j \geq -1/2$ if $(i,j) \notin E$.



Every 3-colorable graph is vector 3-colorable.

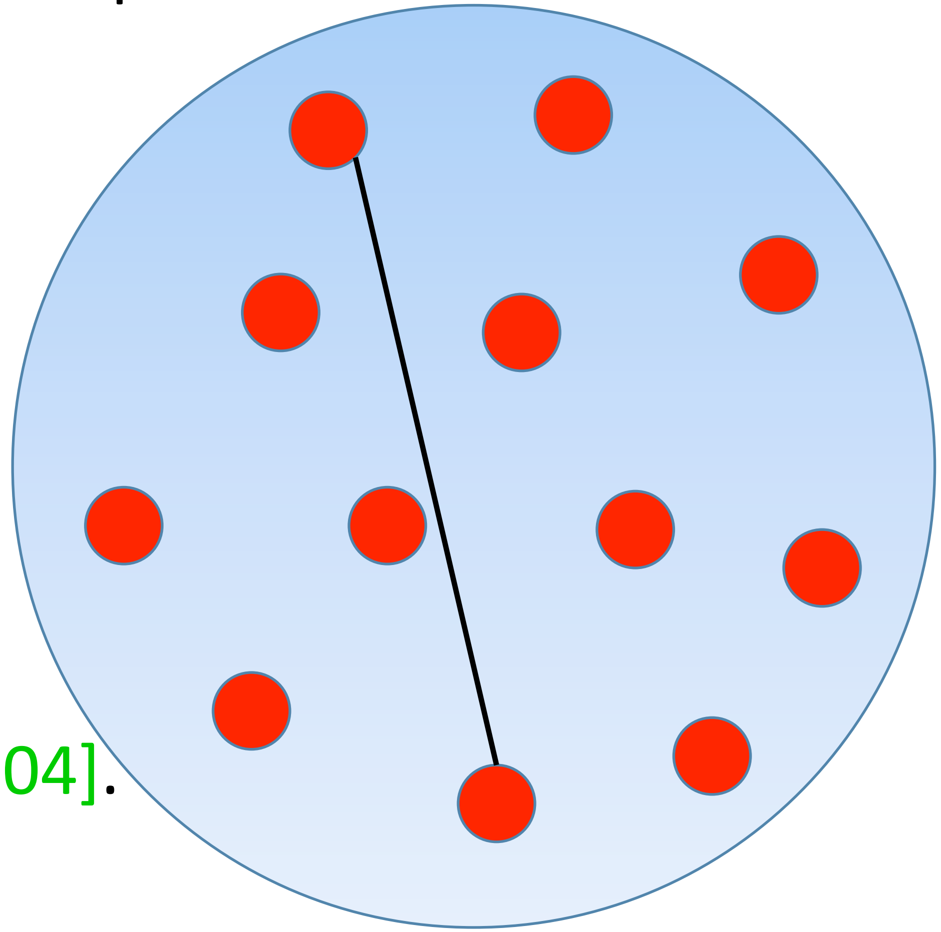
SDP finds a vector 3-coloring in polynomial time.

Anti-geometric graphs

- n vertices placed on a dim-dimensional sphere.
- Edges connect vertices that are far apart (inner angle above $2\pi/3$).

Vector 3-colorable.

Chromatic number roughly $d^{1/3}$ (if vertices evenly spaced).



[Feige, Langberg, Schechtman 2002, 2004].

Number of colors used expressed as n^{δ}

Wigderson 1982, 1983:	0.5
Blum 1989, 1990, 1994:	0.375
Karger, Motwani, Sudan 1994, 1998:	0.25
Blum, Karger 1997:	0.214
Arora, Chlamtac, Charikar 2006:	0.211
Chlamtac 2007:	0.207
Kawarabayashi, Thorup 2014, 2017:	0.199

None of the above improve over $d^{1/3}$

Max 3-coloring

Given a 3-colorable graph on n vertices, 3-color many vertices legally.

- Min 3-coloring with k colors implies $3/k$ approximation to max 3-coloring.
- ρ approximation algorithm for max 3-coloring implies min 3-coloring with $O(\log n/\rho)$ colors (and $O(1/\rho)$ if ρ improves as n decreases).

Known min 3-coloring approximation algorithms are derived from max 3-coloring algorithms.

Remark: for random input instances, a good approximation for max 3-coloring might not imply a good approximation for min 3-coloring.

The random planted 3-coloring model

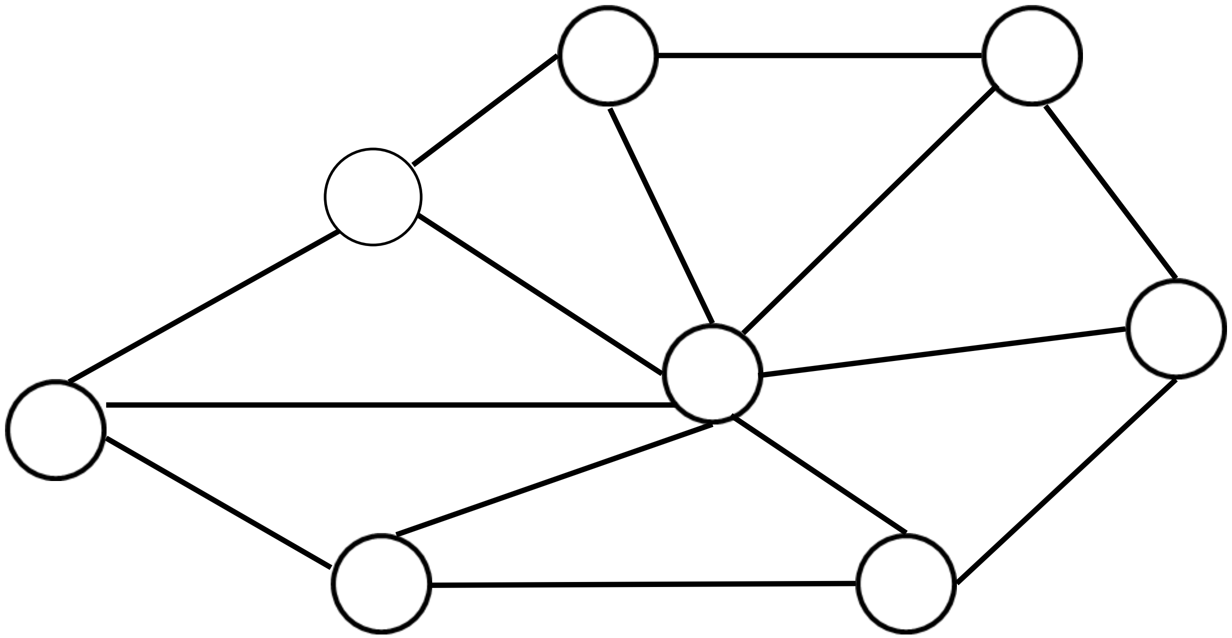
The $G_{n,p,3}$ model of random 3-colorable graphs introduced by Kucera [1977].

An alternative presentation:

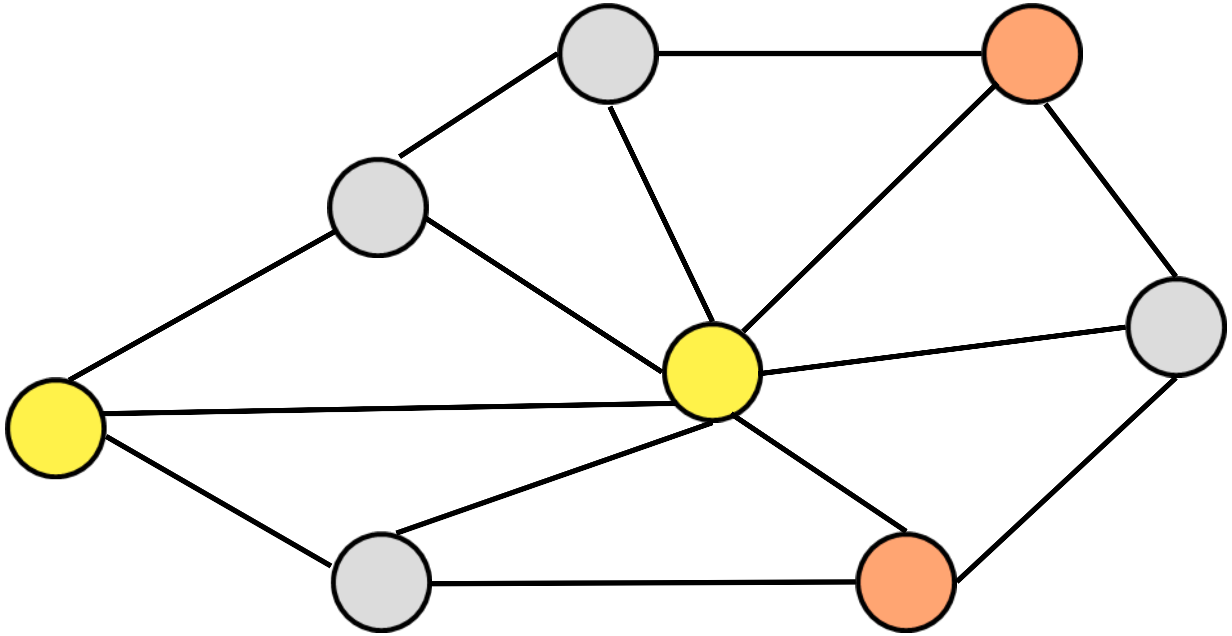
- Start with host graph H sampled from $G_{n,p}$.
- Plant a random 3-coloring P.
- Remove monochromatic edges.

$d=p(n-1)$ is the expected average degree (before planting).

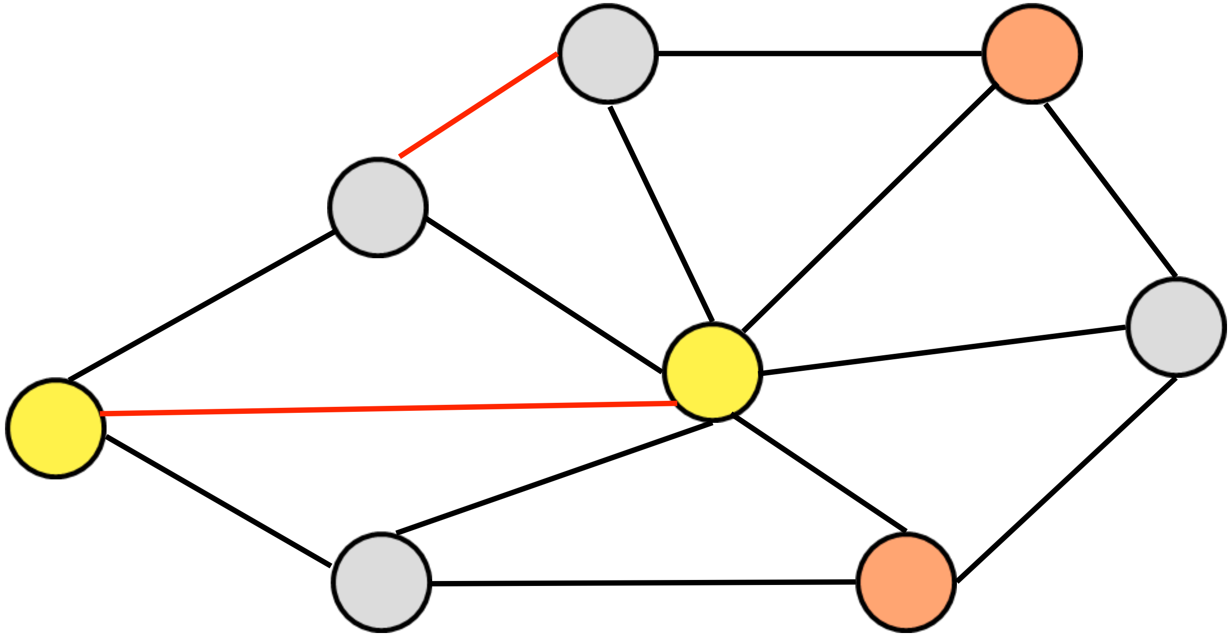
Random host graph H



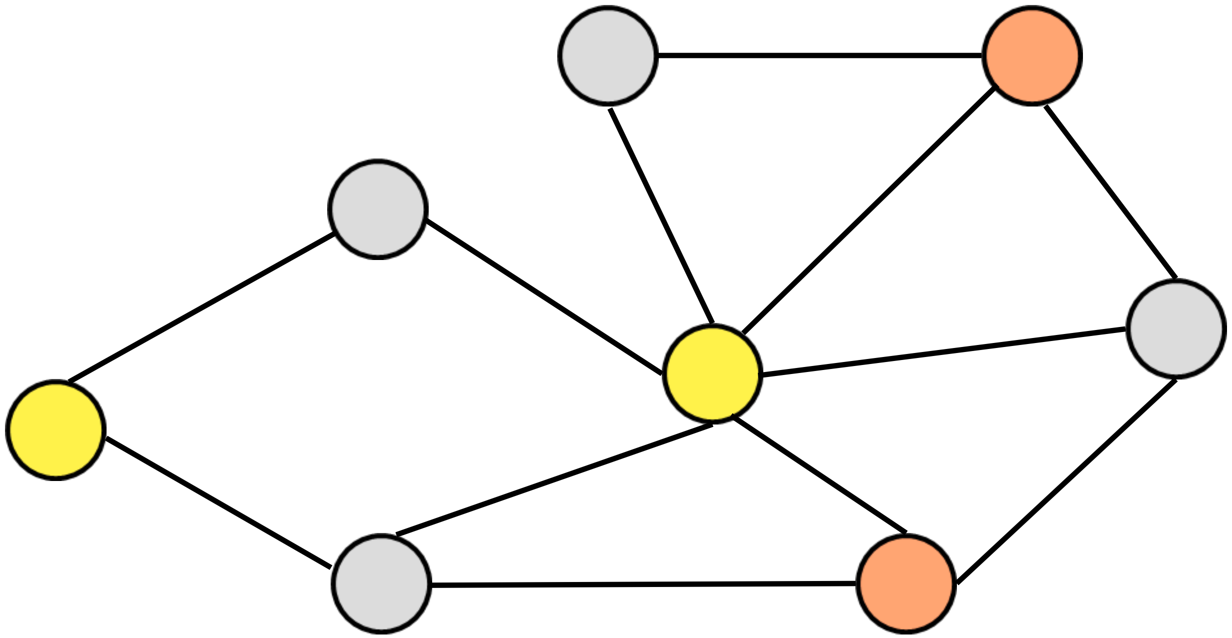
Planted 3-coloring P



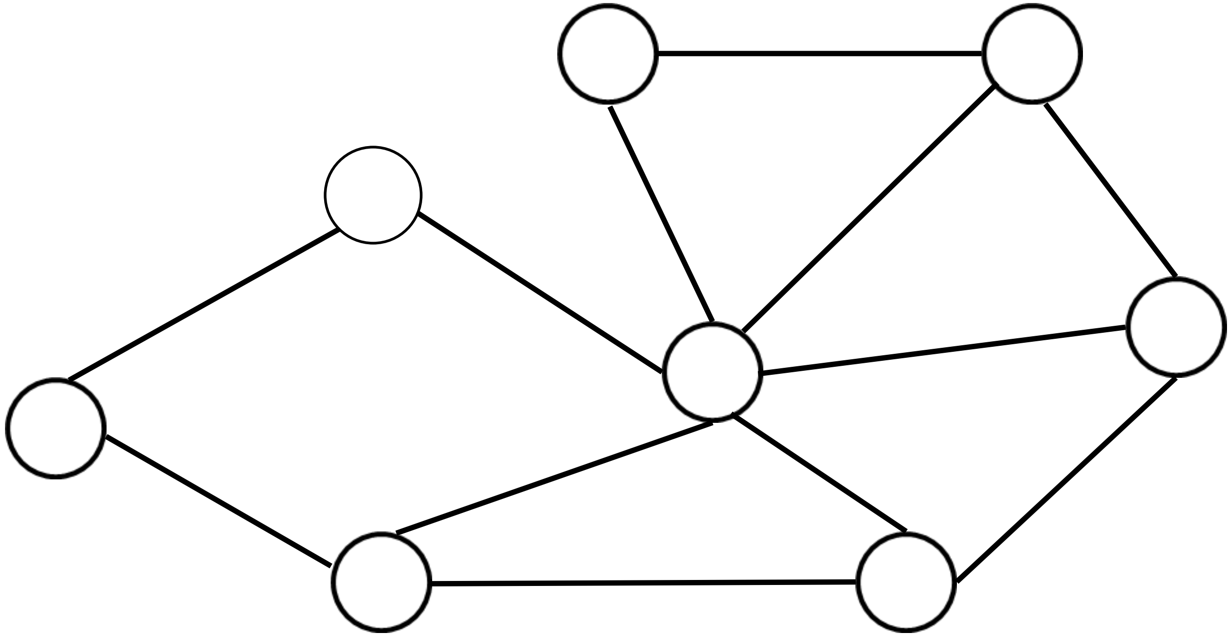
Illegal - monochromatic edges



Remove monochromatic edges



Remove colors →G



The algorithmic task

The input is the graph G.

(The algorithm never sees H or P.)

Task: Find a legal 3-coloring.

G may have several legal 3-colorings. There is no requirement to recover the planted 3-coloring P.

Random 3-colorable graphs

At sufficiently high edge density, a random 3-colorable graph is distributed like a random graph with a planted random 3-coloring.

Such graphs can be 3-colored (w.h.p.) using a spectral algorithm [Alon, Kahale 1994, 1997], and likewise using SDP.

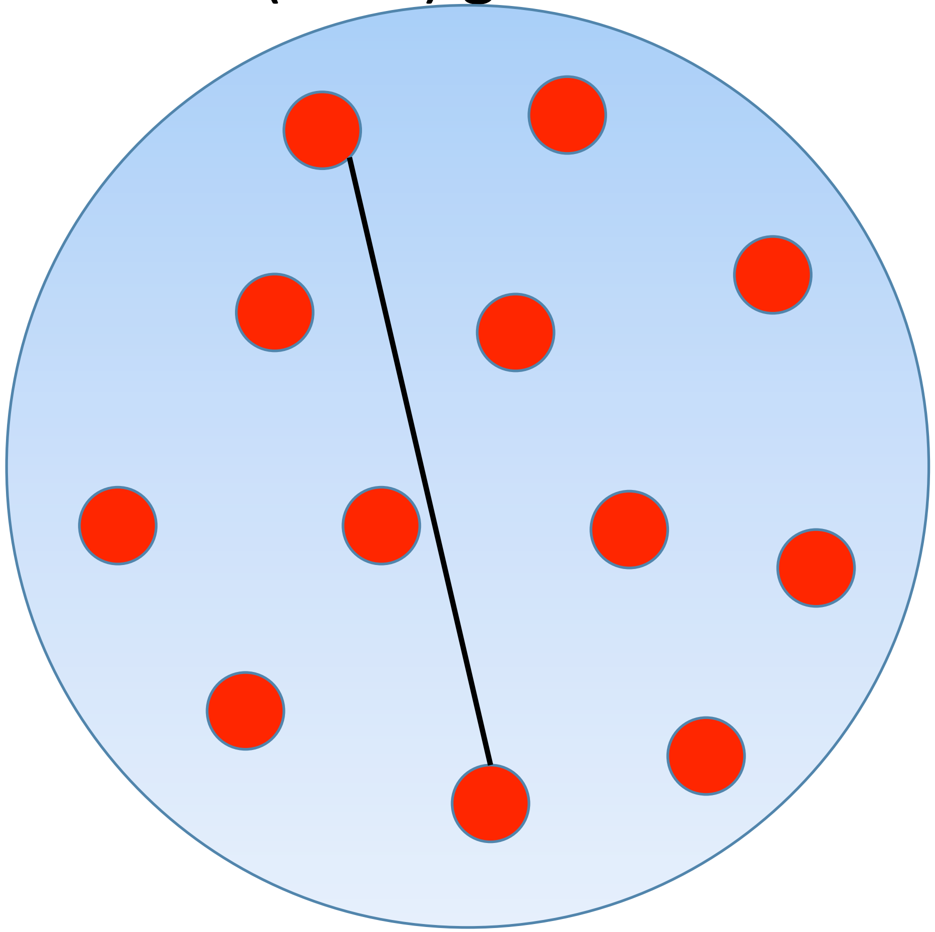
In fact, planted model can be 3-colored even at lower densities (large constant average degree).

Random instances do not seem to capture the difficulties of worst case instances: the known algorithms perform much better on random instances.

A geometric random 3-colorable graph model

The host graph H is a random high dimensional (anti-) geometric graph:

- n vertices are scattered at random on a dim-dimensional sphere.
- Edges connect vertices that are far apart (inner angle above $2\pi/3$).



Plant a random 3-coloring.

(Monochromatic edges then removed.)

A challenge

The input is a graph G generated as above (given as an adjacency matrix, not as an embedding on a sphere).

A legal 3-coloring can be found in polynomial time, when $dim < 4.9326 \log n$ [Roee David, MSc thesis, 2012], corresponding to $\Delta < n^{0.3}$. (At this dimension, a geometric graph supports geometric routing.)

Design an algorithm that works for all dimensions.

The difficulty – the host graph H admits a vector 3-coloring.

(Several candidate algorithms exist – challenges in the analysis.)

A geometric question

Anti-geometric H admits a vector 3-coloring:

v_i are unit vectors

$v_i v_j \leq -1/2$ if $(i,j) \in E$.

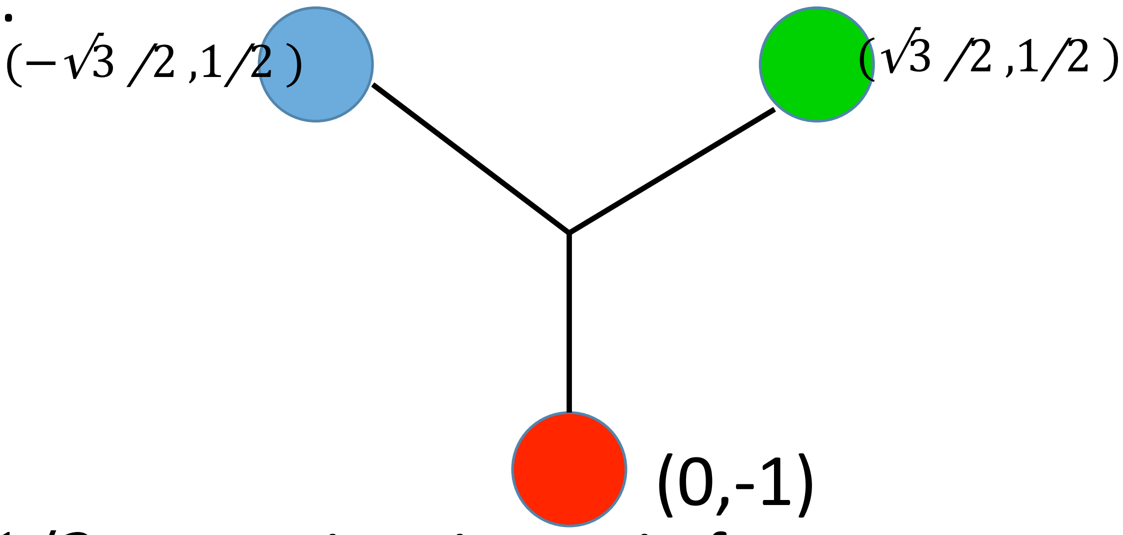
$v_i v_j \geq -1/2$ if $(i,j) \notin E$.

Does it admit a *strong* vector 3-coloring:

$v_i v_j = -1/2$ if $(i,j) \in E$?

If not, may open the way to improve the $d^{1/3}$ approximation ratio for min 3-coloring.

At best, to d^{ε} [Charikar 2002].



Unique games

Graph $G=(V,E)$, k colors, a set of permutations $\pi_{u,v}$ on $[k]$.

Color V so as to maximize the number of legally colored edges.

An edge (u,v) is legally colored if $c(v)=\pi_{u,v}[c(u)]$.

UGC [Khot 2002]: for every $\varepsilon>0$ and $\delta>0$, for sufficiently large k, it is NP-hard to distinguish between instances that are at least $1-\varepsilon$ satisfiable and instances that are at most δ satisfiable.

Random instances

Extensive research on UGC and on its implications (too much to mention).

Random instances of unique games are approximable better than UGC.

In fact, a much stronger statement holds:

Arora, Khot, Kolla, Steurer, Tulsiani, Vishnoi 2008: Unique games on expanding constraint graphs are easy.
Kolla, Makarychev, Makarychev 2011: How to Play Unique Games Against a Semi-random Adversary: Study of Semi-random Models of Unique Games.

Four semi-random models for unique games

Generate a $1-\varepsilon$ satisfiable instance by selecting:

- The graph $G(V,E)$.
- Permutations $\pi_{u,v}$ so that the instance is satisfiable.
- A set E_ε of edges to corrupt.
- The permutations $\pi'_{u,v}$ for the corrupted edges.

Theorem: for sufficiently small $\varepsilon>0$, if at least one of the above selections is made at random (and the other three can be adversarial), then there is a (randomized) polynomial time coloring algorithm for which most edges are legally colored.

(The algorithm requires average degree above $\log k$.)

Dense k-subgraph

- Graph G on n vertices, and parameter k.
- Find subgraph induced on k vertices, of highest average degree.

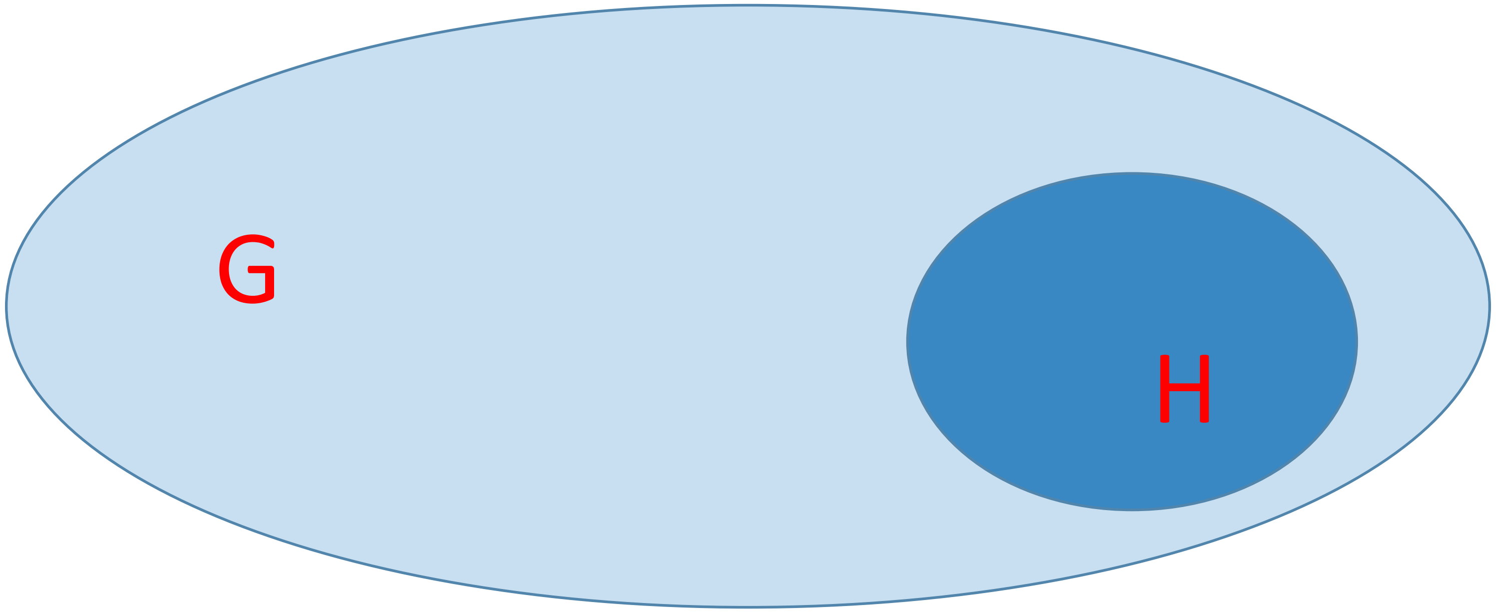
NP-hard (generalizes k-clique).

- Best approximation ratios of the form $n \uparrow \delta$.
- Currently, approximation within a ratio of 2 in quasi-polynomial time is not ruled out.

Random model

$H = G_{k,q}$ planted in $G = G_{n,p}$.

H is densest k-subgraph if $q > p$.



Log-density

Generative model: $H = G_{k,q}$ planted in $G = G_{n,p}$. $q > p$.

If average degree in H is larger than $k^{1/3}$ and average degree in G is smaller than $n^{1/3}$, then H will have cliques of size 4, but G will not.

Can detect existence of H if $\log_k (qk) > \log_n (pn)$ because H will have small induced subgraphs that G does not.

E.g., K_4 at log-density $> 1/3$.

Bhaskara, Charikar, Chlamtac, Feige, Vijayaraghavan: Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. 2010

Generative model: $H = G_{k,q}$ planted in $G = G_{n,p}$. $q > p$.

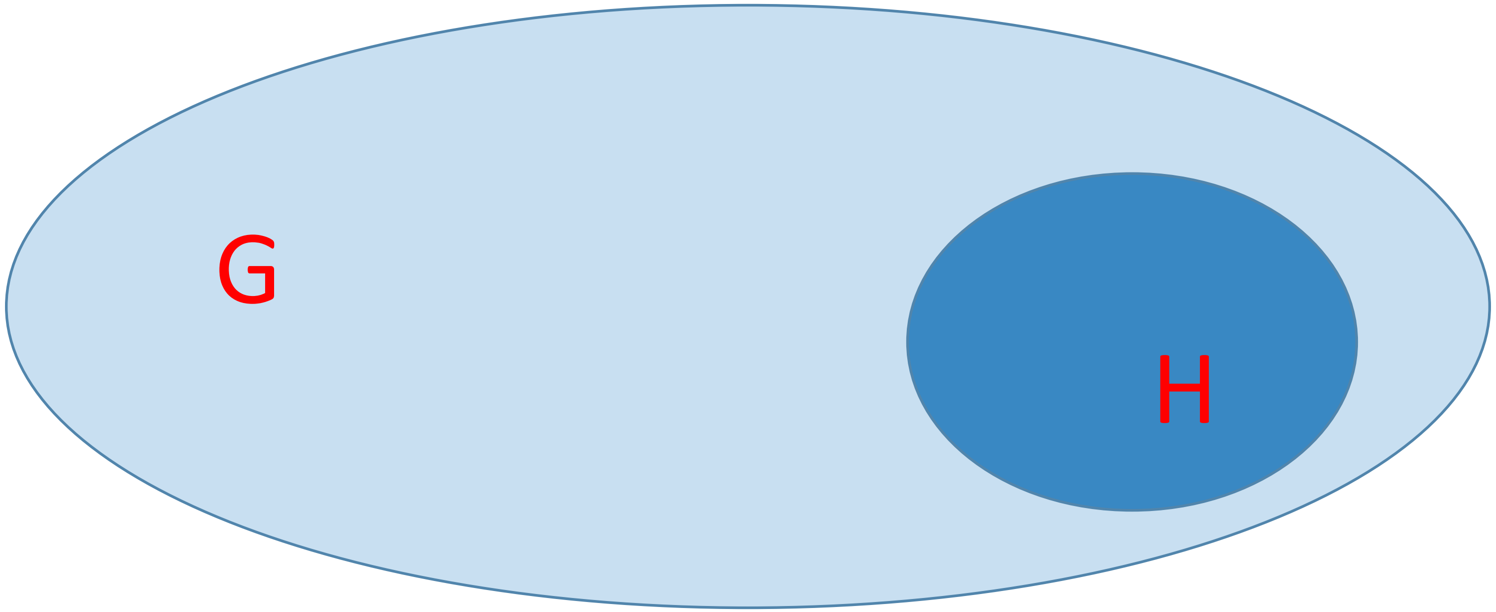
Can detect existence of H if $\log_k(qk) > \log_n(pn)$ because H will have small induced subgraphs that G does not. (E.g., K_4 at log-density $> 1/3$.)

The use of log density was a key insight that led to improved (worst case) $\sim O(n^{1/4})$ approximation ratio for dense k-subgraph.

Open question

$H = G_{k,q}$ planted in $G = G_{n,p}$.

$pn = n^{0.49}$ $\quad$ $k = n^{0.5}$ $\quad$ $qk = k^{0.48}$



Summary

- Max 3-SAT: random instances appear to be as hard as worst case.
- Min bisection: random instances are easy.
- Min 3-coloring: random instances are easy. There are interesting research directions concerning random anti-geometric graphs.
- Unique games: even semi-random (and quarter-random) instances are easy.
- Dense k-subgraph: previous progress inspired by random instances. Current obstacle for further progress manifested by random instances.