Independent sets and graph coloring

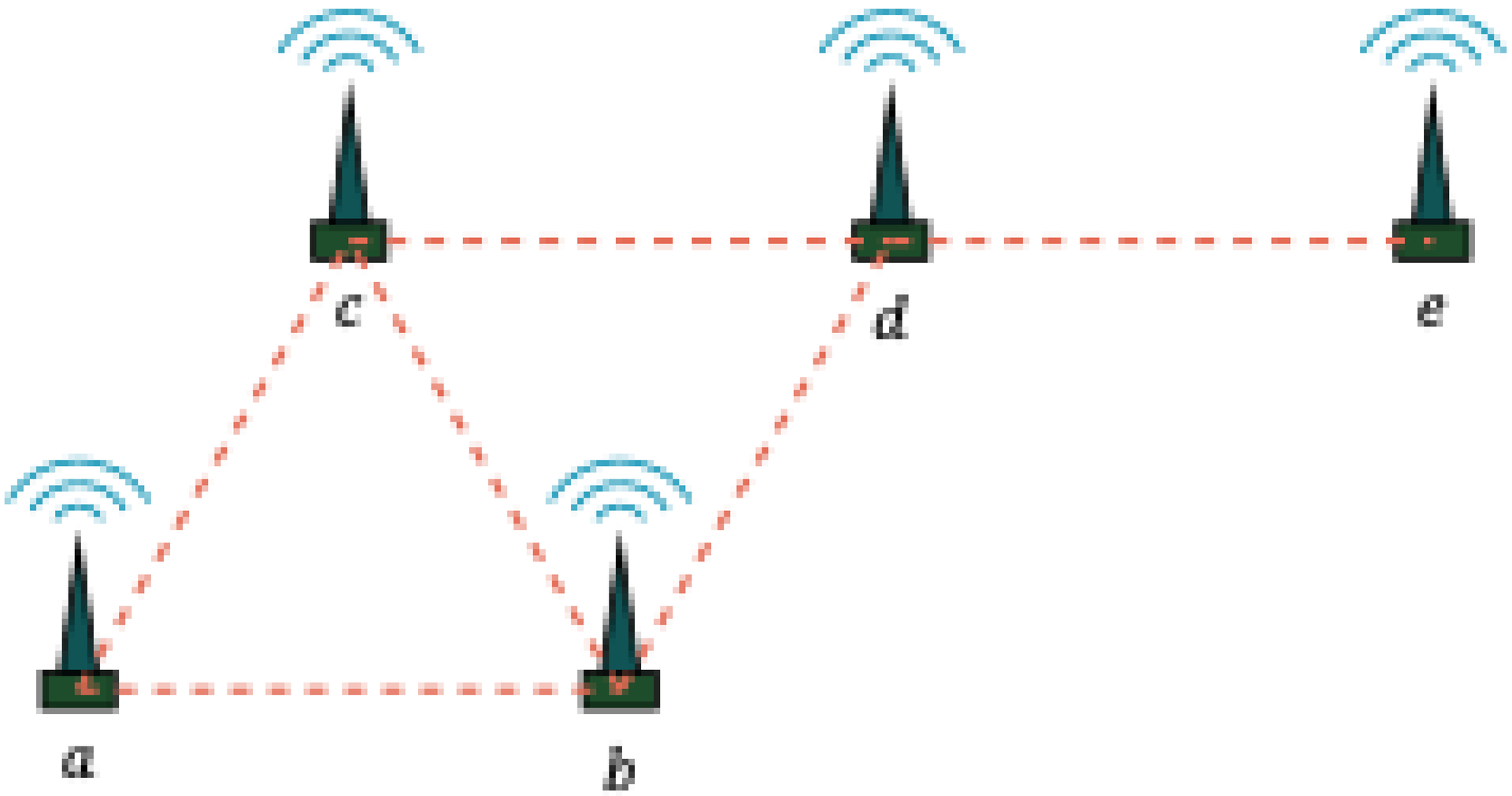
Greedy and randomized greedy algorithms

Definition: graph

- Recall that a **graph** $G = (V, E)$ has a set V of **vertices**, and set E of **edges** where each $e \in E$ is some **unordered pair** of vertices from V
- In computer science and mathematics, many different kinds of graph exist (directed graphs, multigraphs, …)
- In this class we focus on **simple**, **undirected graphs**: there are no loops or multiple edges, and edges are unordered. We also assume the vertex set is **finite**
- In math notation, let $\binom{V}{2} = \{e \subseteq V : |e| = 2\}$ be the set of subsets of V of size 2 and then we require that $E \subseteq \binom{V}{2}$
- Then each edge looks like $\{u, v\}$ where $u, v \in V$ but this is verbose, so we write uv instead

Example

- Consider a small wireless network in which there is a set V of devices that wish to transmit on the network
- If two devices are close together (depending on their transmission power) then when they transmit on the same frequency the transmissions **interfere** and data is lost
- We can model with with a graph: let E be the set of pairs of devices that interfere
- Problems we wish to solve:
  - Maximize the **throughput**: the amount of data we can transmit in one time slot
  - Minimize the **latency**: the length of time a device must wait for some clear frequency between transmissions


c
d
e
a
b

Problem 1: throughput

- In a graph $G = (V, E)$, a set $S \subseteq V$ is called **independent** if there is no edge with both ends in S
- The devices that can transmit at the same time must form an independent set
- Maximizing the throughput corresponds to finding a maximum independent set

Problem 2: latency

- In TDMA, the network schedule considers discrete time slots $1,2,3,\ldots$
- We partition the nodes into independent sets $S_1, \ldots S_k$
- The nodes in S_i are allowed to transmit when the time slot is equal to i modulo k
- The latency is k time slots
- Therefore, we want to minimize the number of parts in a **partition** of the vertices into independent sets

Notation

- We write $\alpha(G)$ for the size of a largest independent set in G
- A **proper coloring** of a graph $G = (V, E)$ is a function $f: V \to \mathbb{N}$ such that for all edges $uv \in E$ we have $f(u) \neq f(v)$
- That is, we think of $\mathbb{N}$ as a set of **colors**, and the coloring is a function giving each vertex a color such that no edge has both its ends the same color
- The **chromatic number** $\chi(G)$ is the least number of colors k for which a proper coloring of G using k colors exists
- This is equivalent to partitioning the V into independent sets, we simply enumerate the partition $S_1, \ldots, S_k$ and let $f(v) = i$ when $v \in S_i$

Examples

Independent sets

- Finding independent sets can be easier than coloring graphs as we only need to make a binary decision for each vertex: it must be in or out of the independent set
- If you're good at finding (large) independent sets then you can do graph coloring reasonably well:

```
c = 0
while G has vertices:
    find an independent set S
    give S color c
    remove S from G
    increment c
```

- We will focus for now on problem 1: finding large independent sets

Examples

Greedy independent set

- Fix some ordering of the vertices: $V = v_1, v_2, \ldots, v_n$ and let $S = \emptyset$
- Process vertices in order
- When processing v_i, put it into S if and only if none of its neighbors are already in S
- How big is S? It depends on the ordering!
- Here are some ideas we'll explore:
  - There is an ordering such that this algorithm finds a largest possible S
  - There are graphs with n vertices and $\alpha(G) = \Omega(n)$ such that some ordering only finds $|S| = 2$
  - We can order the vertices randomly and look at the expected/average size of S

Graph notation

- $N(u) = \{v \in V : uv \in E\}$ is the set of **neighbors** of a vertex u
- $d(v) = |N(v)|$ is the **degree** of a vertex v
- $\delta(G)$ is the **minimum degree** and $\Delta(G)$ is the **maximum degree** over the vertices in G
- A **subgraph** of G is any graph obtained from G by deleting edges and vertices

Greedy independent set performance

Theorem: For all graphs G there is some ordering of the vertices such that greedy gives an independent set of size $\alpha(G)$

Proof: Let I be any independent set. Put I first in the ordering. Then the set S output by greedy contains I. □

Greedy independent set performance

Theorem: For all even $n \geq 2$ there is a graph G on n vertices with $\alpha(G) \geq \frac{n}{2}$ but some ordering of the vertices forces greedy to find a set S of size 2.

Proof: Let $n = 2k$ and let $V = \{u_1, \dots, u_k\} \cup \{v_1, \dots, v_k\}$. Let the edges be given by the rule that for all i, u_i is connected to v_j for all $j \neq i$.

□

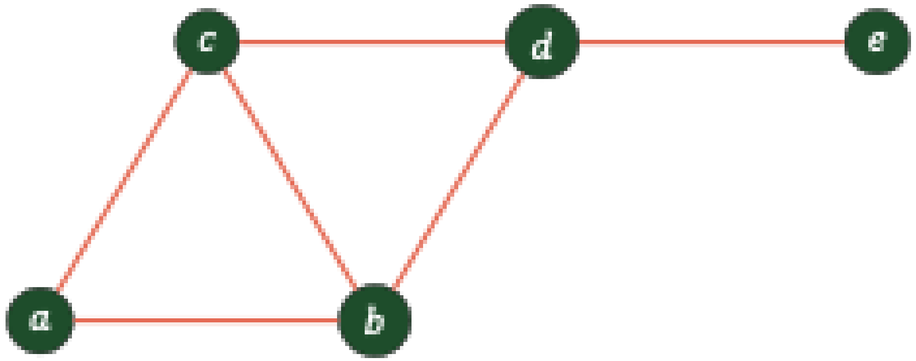
Greedy independent set performance

- We've learned that the performance of greedy is somewhere between **perfect** and **about as bad as possible**
- Let's try an important idea: **make decisions uniformly at random**
- For greedy independent sets this can look like choosing the vertex ordering uniformly at random
- Now the size of S is a random variable, but we can easily study its **expectation**

Discrete probability

- A **probability space** is a finite **sample space** Ω (set of outcomes) and a **probability measure** $\mathbb{P} : \Omega \to [0,1]$
- For all $\omega \in \Omega$ we need $\mathbb{P}(\omega) \in [0,1]$ and $\mathbb{P}(\Omega) = \sum_{\omega \in \Omega} \mathbb{P}(\omega) = 1$
- An **event** is a subset $A \subseteq \Omega$
- A **random variable** is a function $X : \Omega \to \mathbb{R}$, its **expectation** is $\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega)\mathbb{P}(\omega)$
- Expectation is **linear**: $\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$ (where $a, b \in \mathbb{R}$ are constants and X, Y are random variables)
- Expectation is **always linear**: it doesn't matter if X and Y **depend on each other**
- See the probability cheat sheet for computer scientists on Canvas

Define a random independent set in this graph



- $\Omega = \{\, \emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, d\}, \{a, e\}, \{b, e\}, \{c, e\} \,\}$
- Set $\mathbb{P}(\{a, d\}) = \mathbb{P}(\{a, e\}) = \mathbb{P}(\{b, e\}) = \mathbb{P}(\{c, e\}) = 1/4$, the rest 0
- $a \in S$ is an event corresponding to $\{\{a\}, \{a, d\}, \{a, e\}\} \subseteq \Omega$

- $\mathbb{P}(a \in S) =$

- $\mathbb{E}|S| =$

Random greedy independent sets

Theorem: Let $G = (V, E)$ be a graph and let S be the (random) independent set output by the greedy algorithm with the vertices ordered uniformly at random. Then $\mathbb{E}|S| \geq \sum_{v \in V} \frac{1}{d(v)+1}$.

Proof: Let X_v be the indicator random variable for the event that $v \in S$: define $X_v = 1$ if $v \in S$ and $X_v = 0$ otherwise.

Then $|S| = \sum_{v \in V} X_v$ and hence by linearity of expectation $\mathbb{E}|S| = \sum_{v \in V} \mathbb{E}X_v$.

But indicator random variables are special: $\mathbb{E}X_v = \mathbb{P}(v \in S)$.

When is $v \in S$? This definitely holds if v is earlier in the order than any of $N(v)$. So $\mathbb{E}X_v \geq \mathbb{P}\big(v \text{ comes before all of } N(v)\big)$

Every ordering is equally likely, meaning each vertex is equally likely to be in any specific position. So $\mathbb{E}X_v \geq \frac{1}{d(v)+1}$. □

Theorem (Caro–Wei): For all graphs G, $\alpha(G) \geq \sum_{v \in V} \frac{1}{d(v)+1}$.

Random greedy independent sets

How good is the bound?

- For all n, find a graph on n vertices such that $\alpha(G) = \sum_v \frac{1}{d(v)+1}$
- Hint: think of a graph with all the degrees the same first
- What is special about your graph?
- Do you think the randomized greedy algorithm does better if the graph looks very different?

Triangle-free graphs

- A graph is **triangle-free** if it contains no triangles
- In a triangle-free graph, for every vertex v the set $N(v)$ must be independent
- You might think we can find larger independent sets than the Caro-Wei bound in triangle-free graphs
- Let's prove something like this!
- For simplicity, we study triangle-free graphs with given **average degree** instead of looking at the individual degrees in detail. We also just state a bound for $\alpha(G)$ instead of analyzing the randomized greedy algorithm directly
- The proof does bound the expectation of the randomized greedy algorithm, and the method can be extended to give a bound in terms of individual degrees. We avoid those difficulties in CS 420.

Convexity

- A function $f: A \to \mathbb{R}$ is **convex** if for all $x, y \in \mathrm{A}$ and $t \in [0,1]$ we have $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$
- If the second derivative exists on A then f is convex if and only if $f''(x) \geq 0$
- Convex functions lie above their tangent lines:

$$f(y) \geq f(x) + (y-x)f'(x)$$

- A proof of this uses a fancy version of Taylor expansion: if the second derivative exists then for some $c \in [x, y]$,

$$f(y) = f(x) + (y-x)f'(x) + \frac{1}{2}(y-x)^2 f''(c)$$

- **Jensen's inequality**: if f is convex then for any random variable X, $f(\mathbb{E}X) \leq \mathbb{E}f(X)$

Examples

Average degree, Caro–Wei, Túran

Theorem (Caro–Wei): For all graphs G, $\alpha(G) \geq \sum_{v \in V} \frac{1}{d(v)+1}$.

- The **average degree** of a graph on n verties is $\frac{1}{n}\sum_{v \in V} d(v)$, and we denote this $d(G)$
- Note that if G has n vertices then $\frac{nd(G)}{2}$ is the number of edges of G (the **handshake theorem**)
- Consider $f(x) = \frac{1}{x+1}$. Then $f''(x) = \frac{2}{(x+1)^3}$ which is ≥ 0 if $x \geq 0$
- Jensen's inequality now tells us that $\sum_{v \in V} \frac{1}{d(v)+1} \geq \frac{n}{d(G)+1}$
- The Caro–Wei theorem implies $\alpha(G) \geq \frac{n}{d(G)+1}$, a result attributed to Túran

Random greedy without triangles

Theorem (Shearer): Let $f:[0,\infty) \to (0,1]$ be the function with $f(d) = \frac{d\log(d)-d+1}{(d-1)^2}$ and $f(0) = 1, f(1) = 1/2$. Then for any triangle-free graph G on n vertices we have $\alpha(G) \geq n\, f(d(G))$.

As $d \to \infty$ we have $f(d) \sim \frac{\log(d)}{d}$ so we have beaten the Túran bound by a factor $\sim \log(d)$

Lemma: f is continuous on $[0,\infty)$ and for $d \in (0,\infty)$ we have $0 < f(d) < 1$, $f'(d) < 0$, and $f''(d) \geq 0$. The function f also satisfies the differential equation $(d+1)f(d) = 1 + (d-d^2)f'(d)$.

Proof: Homework. □

Since f is convex we have for any d and d' that $f(d') \geq f(d) + (d'-d)f'(d)$

Random greedy without triangles

Theorem (Shearer): For any triangle-free graph G on n vertices we have $\alpha(G) \geq n\, f(d(G))$.

Proof: Induction on n. The result is clear for $n = 1$ because $f(0) = 1$.

For the induction step, we write $d = d(G)$ and note that for any vertex v of G we can form $G' = G - v - N(v)$ and suppose it has n' vertices and average degree d'.

Then G' is triangle-free too, so by induction we must have $\alpha(G) \geq 1 + n' f(d')$. Then we want to show that there's some way of choosing v such that

$$1 + n' f(d') \geq n f(d)$$

By convexity we have $1 + n' f(d') \geq 1 + n' f(d) + n'(d' - d) f'(d)$ so we show this is at least $n f(d)$.

We play the usual trick: **choose v uniformly at random** and show that the inequality holds on average.

Random greedy without triangles

We want to show that when v is uniformly random that

$$\mathbb{E}[1 + n'f(d) + n'(d' - d)f'(d)] \geq nf(d)$$

We compute that $n' = n - d(v) - 1$ and $n'd' = nd - 2\sum_{w \in N(v)} d(w)$.

Note that the computation of $n'd'$ uses that G is triangle-free.

Then it is enough to show that $\mathbb{E}[1 + n'(d' - d)f'(d)] \geq \mathbb{E}[(d(v) + 1)f(d)]$.

$$\mathbb{E}[n'd'] = \mathbb{E}\left[nd - 2\sum_{w \in N(v)} d(w)\right] = nd - \frac{2}{n}\sum_{v \in V}\sum_{w \in N(v)} d(w) = nd - \frac{2}{n}\sum_{w \in V} d(w)^2 \leq nd - 2d^2$$

By the convexity of $x \mapsto x^2$ and Jensen's inequality

Then because $f'(d) < 0$ we have $\mathbb{E}[1 + n'(d' - d)f'(d)] \geq 1 + (d - d^2)f'(d) = (d + 1)f(d)$,

Where the final equality is by the differential equation lemma.

But $\mathbb{E}[(d(v) + 1)f(d) = (d + 1)f(d)$, hence we are done. □

Random greedy without triangles

That was a tricky proof to understand.

We had three main ideas:

- Proof by induction
- Convexity
- The probabilistic method

We also had to do some magic: the function f seems like an inspired choice!

To help digest the proof, highlight in red any steps that use induction, highlight in green any steps that use convexity, and highlight in blue any steps involving the probabilistic method.

References:

- https://doi.org/10.1016/0012-365X(83)90273-X (Shearer's original paper)
- https://doi.org/10.1016/0095-8956(91)90080-4 (A follow-up handling individual degrees)
- https://doi.org/10.48550/arXiv.2503.10002 (A generalization of Shearer's original paper on counting independent sets instead of finding their size)
- https://github.com/Pjotr5/ShearerTriangleFreeInd (A formal proof in Lean 4 of some of these results)

Local fractional coloring

- The proof of the Caro–Wei theorem tells us that in any graph there's a probability distribution on independent sets such that every vertex satisfies $\mathbb{P}(v \in S) \geq \frac{1}{d(v)+1}$
- Martinsson and Steiner (https://doi.org/10.48550/arXiv.2501.00567) generalized Shearer's theorem by proving that in any triangle-free graph there's a probability distribution on independent sets such that every vertex satisfies $\mathbb{P}(v \in S) \geq f(d(v))$, where f is the same magic function from Shearer's theorem
- These results connect to something we won't study called **fractional coloring with local demands**

Greedy coloring

- Label our colors $1,2,3,\ldots$
- Fix some ordering of the vertices: $V = v_1, v_2, \ldots, v_n$
- Color in order, using the least available color at each step
- How many colors do we use? It depends on the ordering!

Graph structure and coloring

Theorem: For all graphs G we have $\chi(G) \leq \Delta(G) + 1$.

Proof: Consider the greedy algorithm. When we come to a vertex v, at worst it has $\Delta(G)$ neighbors that have all got different colors already. But then out of the colors $1,2,\dots,\Delta(G)+1$ there must be an available one for v. □

Homework: for each n think of a graph on n vertices in which this theorem is tight.

The graphs in which this is tight seem to have a lot of triangles, can we do better in a triangle-free graph?

Theorem (Molloy): A triangle-free graph G of maximum degree Δ has $\chi(G) \leq \big(1 + o(1)\big)\frac{\Delta}{\log \Delta}$, and there is an efficient randomized algorithm that finds a coloring certifying this bound.

Note that again we have beaten the greedy bound by a log factor.

Studying Molloy's algorithm has been a major topic in my research…

Random graphs

- So far we have studied greedy algorithms for independent sets and proper colorings
- Based on some assumptions about the input graph (degrees, lack of triangles), we gave bounds on the performance of the algorithm in the **worst case**
- Another important type of algorithm performance is the **average case**
- To compute this, we must decide a probability distribution on the input to the algorithm and study it
- For graphs, one simple option is to study the **binomial random graph** $G(n, p)$
- We fix a number of vertices n and a probability p
- Every one of the possible edges included with probability p, and we make each decision independently

Independence number of $G(n,p)$

Let X_k be the number of independent sets of size k in $G(n,p)$

There are $\binom{n}{k}$ of vertices sets of size k, each is independent with probability $(1-p)^{\binom{k}{2}}$

Fact: $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$

For convenience, let $b = \frac{1}{1-p}$

$$\mathbb{E}X_k = \binom{n}{k} b^{-\binom{k}{2}} \leq \left(\frac{en}{k}\right)^k b^{-\binom{k}{2}} = \left(\frac{en}{k} b^{-\frac{k-1}{2}}\right)^k = \left(\frac{en}{\sqrt{k}} b^{-\frac{k}{2}}\right)^k$$

This gets smaller as k grows, and we study the point at which it crosses 1

When $b^{k/2} \geq n$, or $k \geq 2\log_b n$, it is definitely less than 1

It's less obvious but when $k < (2-\epsilon)\log_b n$ it is significantly more than 1 (for any constant $\epsilon \in (0,2)$ and large enough n)

Independence number of $G(n,p)$

Let $\epsilon > 0$ be small

For $k \geq 2\log_b n$ we have $\mathbb{E}X_k < 1$

For $k \leq (2-\epsilon)\log_b n$ we have $\mathbb{E}X_k > 1$

But this isn't enough to show that it's likely there's an independent set of size $\approx 2\log_b n$

To do that we use **Chebyshev's inequality**

$$\mathbb{P}(X_k = 0) \leq \frac{\mathrm{Var}(X_k)}{\mathbb{E}\,[X_k]^2} = \frac{\mathbb{E}\left[X_k^2\right]}{\mathbb{E}\,[X_k]^2} - 1$$

If we can find a k such that this is very small, then it's likely there really is an independent set of size k.

Annoying math

When $k = (2-\epsilon)\log_b n$,

$$\frac{\mathbb{E}\left[X_k^2\right]}{\mathbb{E}\left[X_k\right]^2} = \sum_{i=0}^{k}(a+b)^n = \sum_{k=0}^{n}\binom{k}{i}\binom{n-k}{k-i}\binom{n}{k}^{-1} b^{\binom{i}{2}} \approx 1$$

Urgh. Let's just convince ourselves we can compute $\mathbb{E}\left[X_k^2\right]$. Let S range over sets of size k and let I_S be the indicator random variable for S being independent

$X_k = \sum_S I_S$ and $X_k^2 = (\sum_S I_S)(\sum_T I_T) = \sum_{S,T} I_S I_T$

To compute the expectation $\mathbb{E}X_k^2$ we need to know, for any two sets of size k, the probability they're both independent.

This depends on their overlap, say i vertices. Then $\mathbb{E}I_S I_T = (1-p)^{2\binom{k}{2}-\binom{i}{2}}$

And there are $\binom{k}{i}\binom{n-k}{k-i}$ pairs of sets of size k that overlap in exactly i vertices…

Theorem: With probability $1 - o(1)$ as $n \to \infty$, we have $\alpha\big(G(n,p)\big) \approx 2\log_b n$

Average case analysis

- We now have a “target” to compare to
- It’s very likely that the largest independent set in $G(n,p)$ has size roughly $2\log_b n$ with $b = \frac{1}{1-p}$
- But how does the greedy algorithm do?

Greedy IS in a random graph

The greedy algorithm is equivalent to choosing a vertex and then deleting it and its neighbors

If we start with n vertices then on average we delete $1 + p(n-1) \approx pn$ vertices and keep $\approx (1-p)n$

After k steps we will have roughly $(1-p)^k n$ vertices left

We stop when there is less than 1 vertex remaining

This will be when $k \approx \log_b n$

Again, by doing tedious math we can make this precise…

Theorem: The greedy algorithm likely finds an independent set of approximately half the maximum size in a binomial random graph.

Open problem: If you can find any efficient algorithm that does better than 1/2 for this problem, you'll get a paper in a top theoretical computer science problem.

More random independent sets

Inspired by physics, we study the following distribution on independent sets:

Fix some $\lambda > 0$ and let $\mathbb{P}(I) = \frac{\lambda^{|I|}}{Z}$

Since probabilities sum to 1, we have $Z = \sum_I \lambda^{|I|}$

It turns out that in any graph on n vertices we have

$$\mathbb{E}|I| \geq \sum_v \frac{\lambda}{1 + (d(v) + 1)\lambda} \geq n \frac{\lambda}{1 + (d + 1)\lambda} \geq n \frac{\lambda}{1 + (\Delta + 1)\lambda}.$$

And in a triangle-free graph we have

$$\mathbb{E}|I| \geq n \frac{\lambda}{1 + \lambda} \frac{W\big(\Delta \log(1 + \lambda)\big)}{\Delta \log(1 + \lambda)},$$

where W is the **Lambert W-function** satisfying $W(x) \sim \log x$ as $x \to \infty$.

Again, we gain roughly a log by excluding triangles. This is the beginning of understanding Molloy's algorithm…