Approximation Algorithms For Projection Games

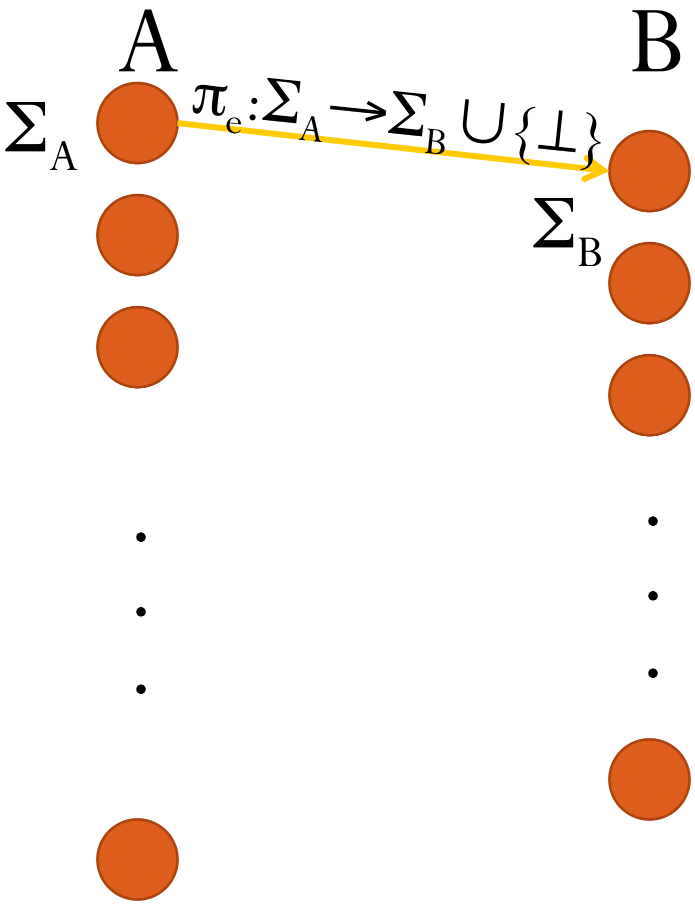
Dana Moshkovitz, MIT

Joint work with Pasin Manurangsi, MIT

"It's good to look at algorithms once in a while as a sanity check on your lower bounds."

Michael Sipser

Projection Games ("Label Cover")

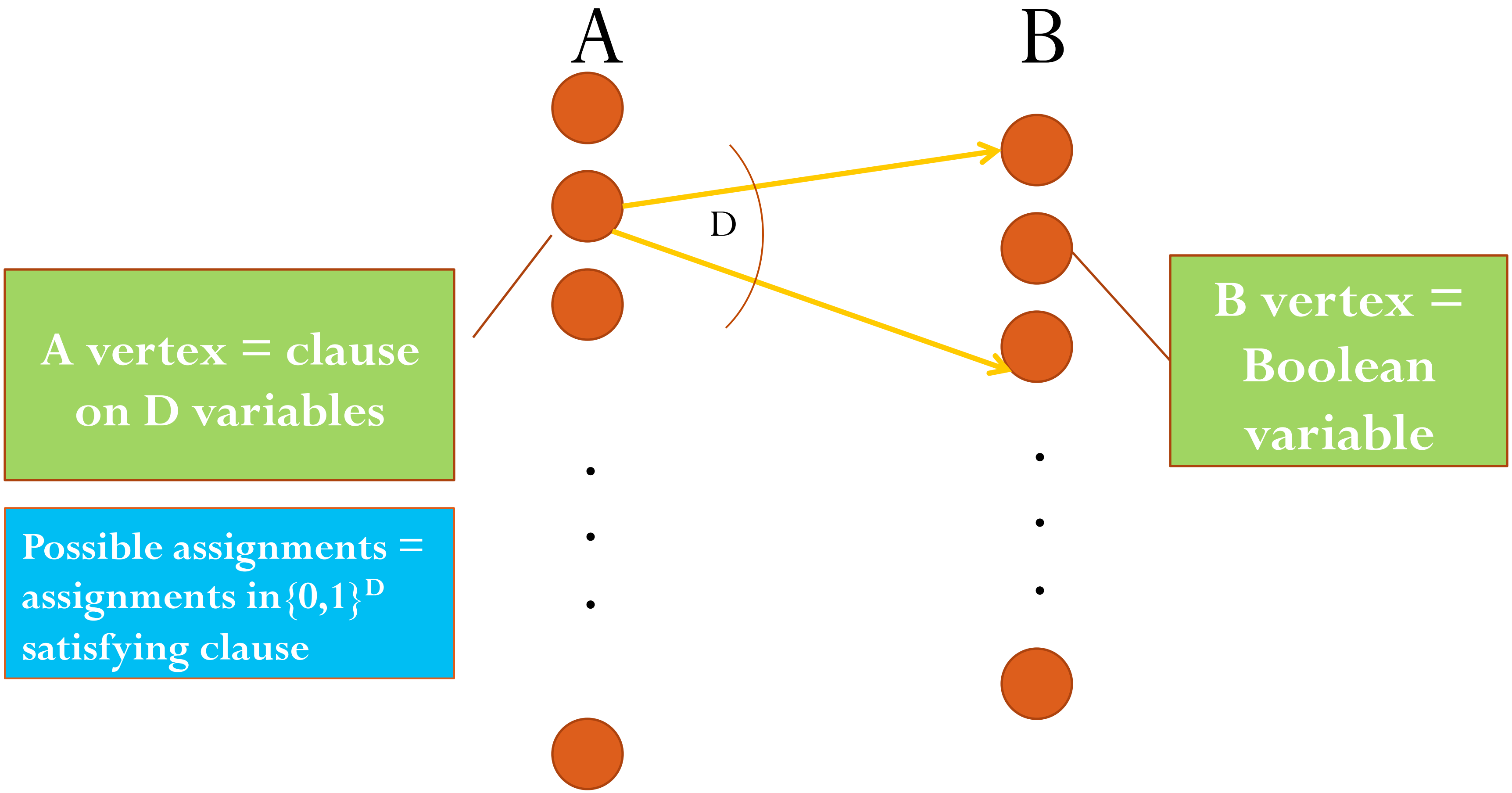


An edge $e=(a,b)\in E$ is satisfied by assignments $f_A:A\to\Sigma_A$, $f_B:B\to\Sigma_B$, if $\pi_e(f_A(a))=f_B(b)$.

Label Cover: Given $G=(G=(A,B,E),\Sigma_A,\Sigma_B, \{\pi_e\}_e)$, Find $f_A:A\to\Sigma_A$, $f_B:B\to\Sigma_B$ maximizing fraction of satisfied edges.

Instance (nearly) *satisfiable* if (almost) all edges can be satisfied.

Example I: D-SAT Game

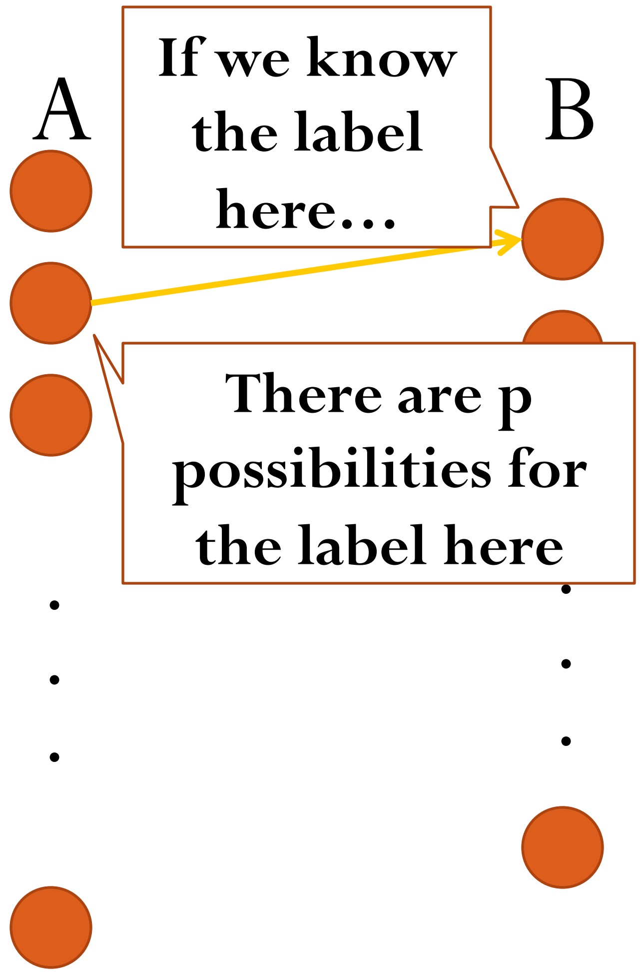


Example II: Unique & p-to-1 Games

- We say that a projection game is **"p to 1"** if

$$p = \mathrm{Max}_{e \in E,\ \sigma \in \Sigma B} \, |\pi_e^{-1}(\sigma)| .$$

- *Unique games* are **1 to 1** games.



This Work

Combinatorial algorithms for satisfiable and nearly satisfiable projection games:

1. Poly-time $\Omega((1/|E||\Sigma_A|)^{1/4})$-approximation for satisfiable projection games.
2. Sub-exponential time exact algorithm for *smooth* satisfiable projection games.
3. PTAS for satisfiable and nearly-satisfiable projection games *on planar graphs*.

Strong PCP Theorem

[Raz94, M-Raz08]

There is $c>0$, such that for any $\varepsilon=\varepsilon(n)\geq 1/n^c$, there is $k=k(\varepsilon)$, such that given a projection game of size n and alphabet size k such that all its edges can be satisfied simultaneously, it is NP-hard to find an assignment that satisfies more than ε fraction of the edges.

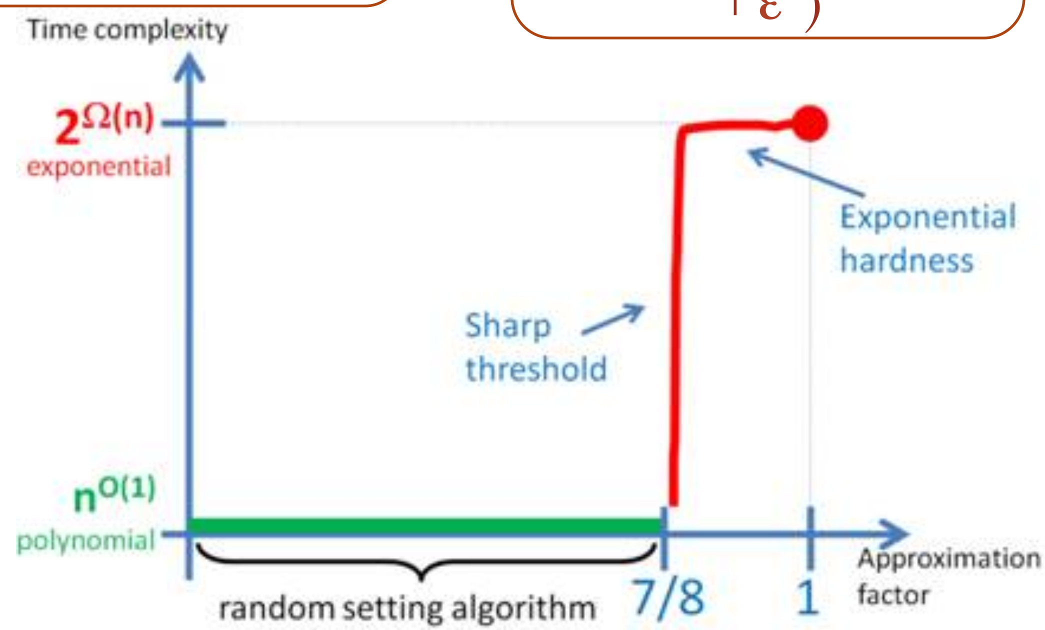
Most optimal NP-hardness of approximation results are based on this theorem…

Hardness of Approximation From Projection Games

- [...,Bellare,Goldreich,Sudan 95, Håstad 97]: *MAX-3SAT* is NP-hard to approximate to within $7/8 + \varepsilon'$.
- ε' is determined by ε of the projection game.

For other problems threshold is 0

For minimization problems: $1/(T + \varepsilon')$



What is the best tradeoff between n, k and ε?

- [Raz 94] (parallel repetition): NP-hard even for $k \leq poly(1/\varepsilon)$ for const $\varepsilon > 0$.
- [M, Raz 08]: $k \leq \exp(poly(1/\varepsilon))$ for any $\varepsilon \geq 1/n^c$.
- [Dinur, Steurer 13] (parallel repetition of [M, Raz 08]): $k \leq \exp(1/\varepsilon)$ for any $\varepsilon \geq 1/n^c$.
- "Projection Games Conjecture": $k \leq poly(1/\varepsilon)$ for any $\varepsilon \geq 1/n^c$.

- Folklore: can satisfy $\varepsilon \geq 1/n, 1/k$ fraction of the edges.
- [Peleg 02]: $\varepsilon \geq 1/(nk)^{1/2}$.
- [Charikar, Hajiaghayi, Karloff 09]: $\varepsilon \geq 1/(nk)^{1/3}$.
- [Manurangsi, M 13]: $\varepsilon \geq 1/(nk)^{1/4}$.

Poly-Time, Poly-Approximation:

- ***Simplifying assumption:*** graph bi-regular; p-to-1 (possibly for a large p).

Overall Approach: Win-Win

- Algorithm 1: Satisfies $1/D_B$ fraction, where D_B=degree of B vertices.
- Algorithm 2: Satisfies $p/|\Sigma_A|$ fraction, where p=number of pre-images of a label in Σ_B.
- Algorithm 3: Satisfies $hD_A/|E|p$ fraction, where h=largest number of neighbors of neighbors of an A vertex.
- Algorithm 4: Satisfies $\Omega(D_B/D_A h)$ fraction.

Approximation factor = max of above four ≥ (multiplication of above four)$^{1/4}$ = $\Omega((1/|E||\Sigma_A|)^{1/4})$.

1. $1/D_B$ Approximation

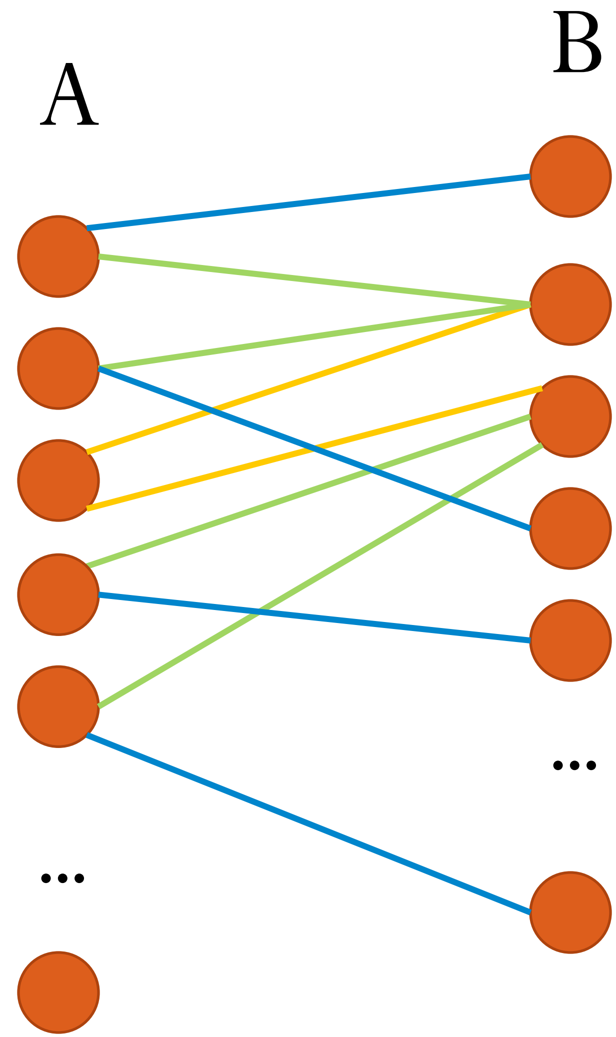
- Pick an arbitrary assignment to the A vertices.
- Per B vertex decide about one neighbor and satisfy the edge between them.

2. $p / |\Sigma_A|$ Approximation

- Pick an assignment at random.
- Can derandomize by a greedy algorithm.

3. $hD_A/p|E|$ Approximation

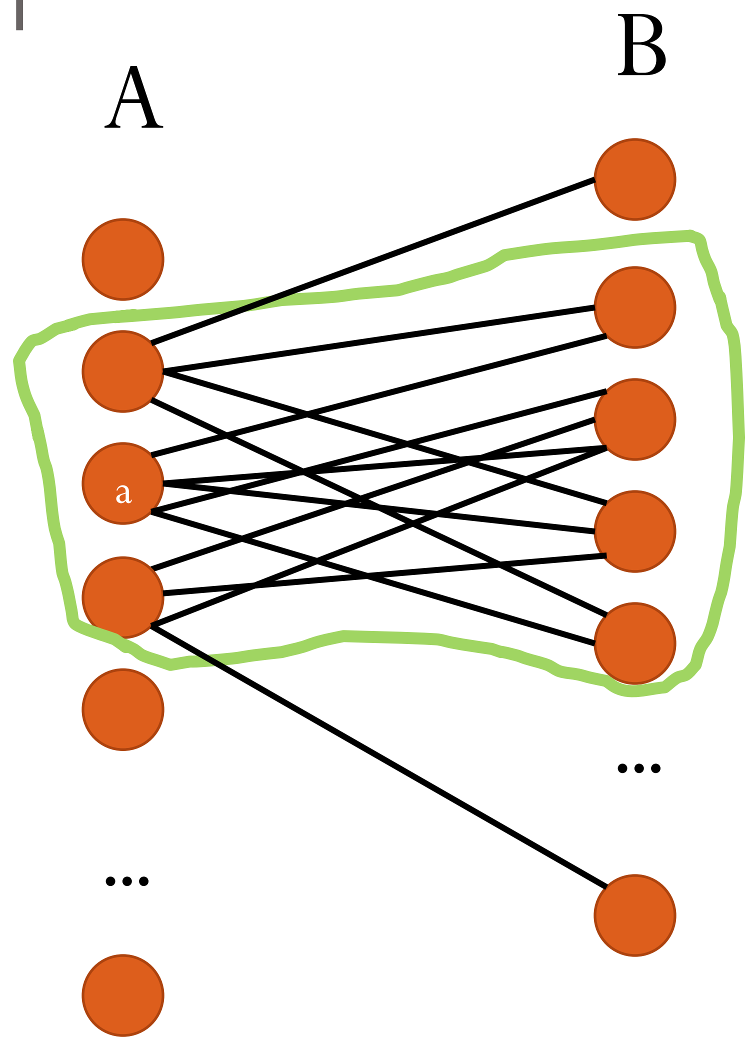
- Let $N(a)$=a's neighbors; $N_2(a)$=a's neighbors' neighbors;
- Go over all possible assignments to a:
  - Get labels for the D_A vertices in $N(a)$.
  - Get p labels for the h vertices in $N_2(a)$.
- There must be an assignment that satisfies hD_A/p edges that touch $N_2(a)$, and we can find it greedily.



4.

$\Omega\ (D_B / D_A\ h)$ Approximation

- Take $a \in A$ such that $|N_2(a)| \leq h$.
- Find an assignment that satisfies all $D_A D_B$ edges on $N(a) \cup N_2(a)$.
- **Claim:** Can continue $\approx |A| / hD_A$ times, each time satisfying $\approx D_A D_B$ new edges.



PTAS for Planar Graphs

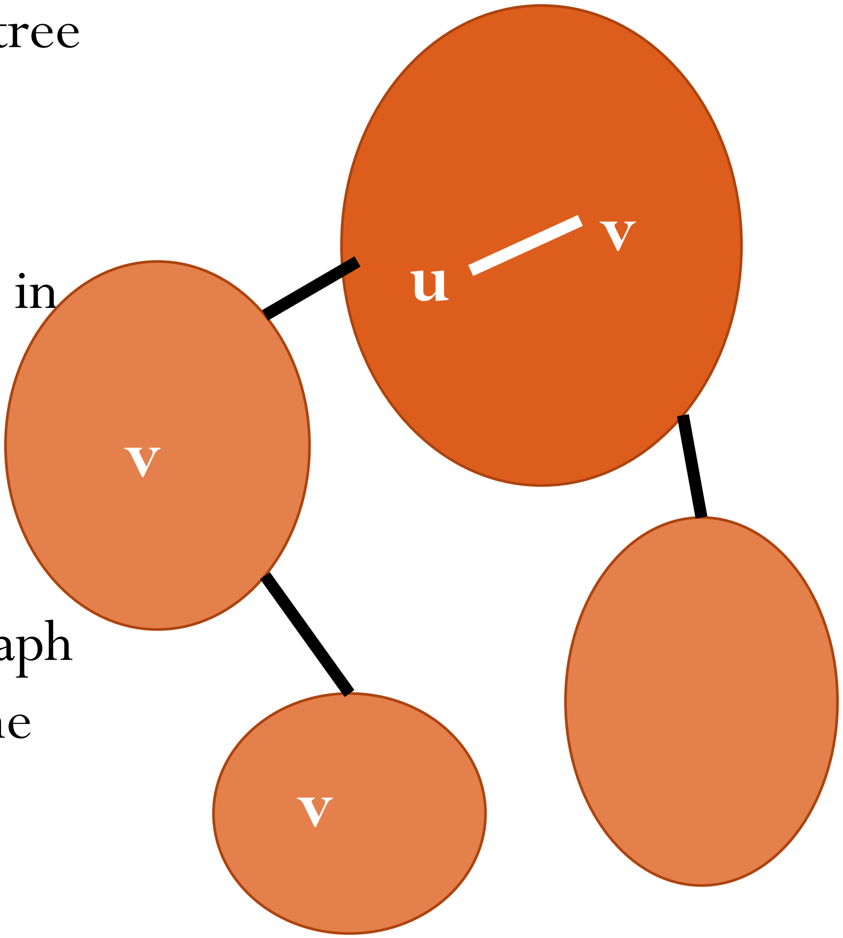
General approach:

1. Delete a few edges to ensure constant tree-width.
2. Solve using dynamic programming.

Tree Decomposition & Tree-Width

- Subsets $B_1, \ldots, B_n$ of vertices and tree on them.
- Every edge is inside some B_i.
- If a vertex is in B_i and B_j, then it's in all B_l's on their tree path.

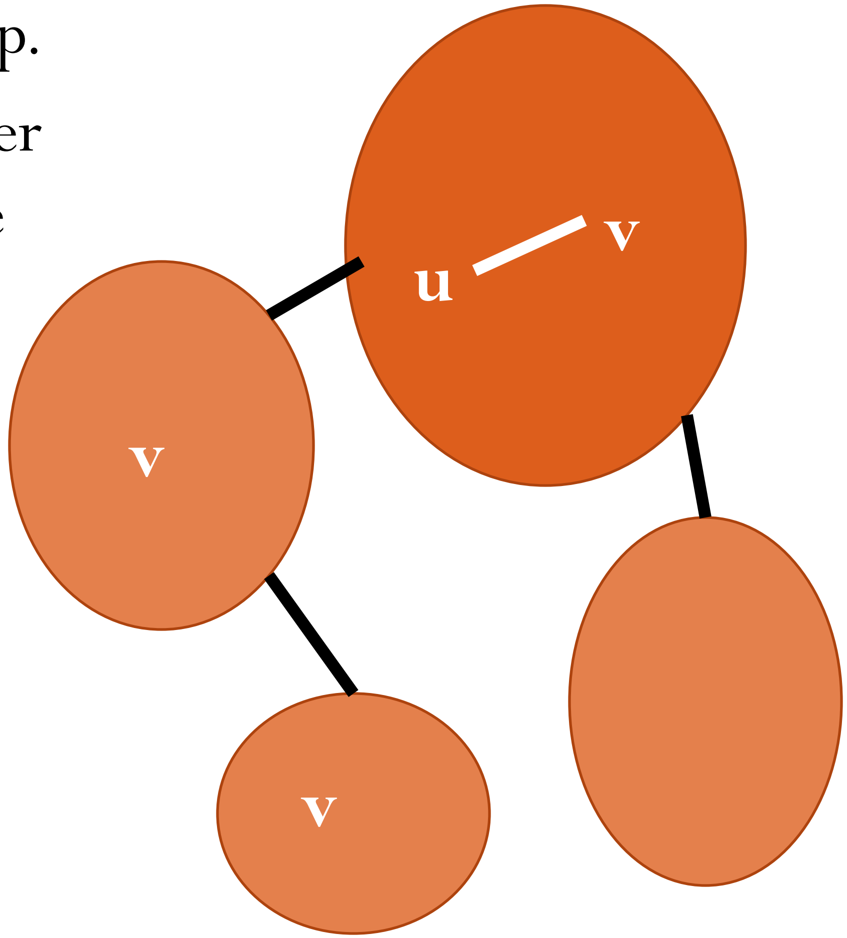
Tree-width = $\max |B_i| - 1$



Theorem (Klein): For any planar graph and number k, can find in linear time at most 1/k fraction of edges to remove, so tree width O(k).

Algorithm For Constant Tree-Width Graphs

- Scan tree on B_i's from leaves up.
- Per assignment inside B_i register how many edges in its sub-tree satisfies.



Saw Two Algorithms:

- Poly time $\Omega((1/|E||\Sigma_A|)^{1/4})$-approximation for satisfiable projection games.
  - What's the right dependence? $(1/|E||\Sigma_A|)^{o(1)}$ would contradict the Projection Games Conjecture.
- PTAS for projection and unique games on planar graphs.
  - More easy projection/unique games?