CSC791/495

Dr. Blair D. Sullivan

September 29, 2017

And you thought this class was hard before....

We've seen "lots" of techniques for designing FPT algorithms, but how do we know when they're unlikely to succeed?

Reasonable questions

- Can we prove a problem is NOT FPT?
- Is problem X as hard as problem Y?
- * Can we guarantee a problem X does NOT have an $2^{o(k)} n^{O(1)}$ algorithm?

Answering these requires two ingredients:

① a notion of reduction that is useful. (if $A \xrightarrow{R} B \Rightarrow B \in FPT \Rightarrow A \in FPT$)

② a hypothesis about distinct complexity classes (e.g. P vs NP)

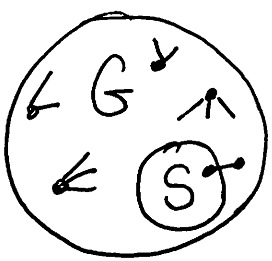
Reductions

Recall A polynomial many-one reduction (aka Karp reduction) is a function that maps instances of problem A to instances of problem B $f: x \to x'$ "from A to B"
s.t. ① f is poly-time computable
② x is a YES-instance of $A \iff f(x)$ is a YES-instance of B.

Implies: If $B \in P \Rightarrow A \in P$. If A is NP-hard $\Rightarrow$ B is NP-hard.

Example Give a reduction from k-IndSet to k'-Vertex Cover

observe: largest independent set is "inversely related" to the smallest vertex cover



$|S| = k$
S independent $\Rightarrow$ u ····· v is not an edge for all $u, v \in S$.

$\Rightarrow$ no edges inside S to cover $\Rightarrow$ we don't need any vertices of S in a cover: $G \setminus S$ is a vertex cover of size $n-k$.

- Is $G \setminus T$ independent for any VC T? YES.

$$\underset{IS}{(G, k)} \to \underset{VC}{(G, n-k)}$$

Parameterized Reductions

Definition A parameterized reduction from a problem A to a problem B is: a function that maps instances (x,k) of A to instances (x',k') of B s.t.

1. parameter preserving: $k' \le g(k)$ for a computable function $g: \mathbb{N} \to \mathbb{N}$
2. answer preserving: (x,k) yes for $A \iff (x',k')$ yes for B.
3. poly-time: runs in $f(k)|x|^{O(1)}$ where f is a computable function.
   $(|x'| \le f(k)|x|^{O(1)})$

Note: you may assume f, g are non-decreasing

Thm: If B is FPT $\Rightarrow$ A is FPT. If A is NOT FPT $\Rightarrow$ B is NOT FPT.

Example: k-IS has a param. reduction to k-Clique.

$(G,k) \longmapsto (\overline{G},k)$

complement

Multicolored Clique

Problem: Given a graph G and a vertex k-coloring, find k vertices so that all have distinct colors and they form a k-clique

Goal show this is exactly as hard as k-CLIQUE.

Observe every k-clique is necessarily multicolored (b/c color classes are indep. sets)

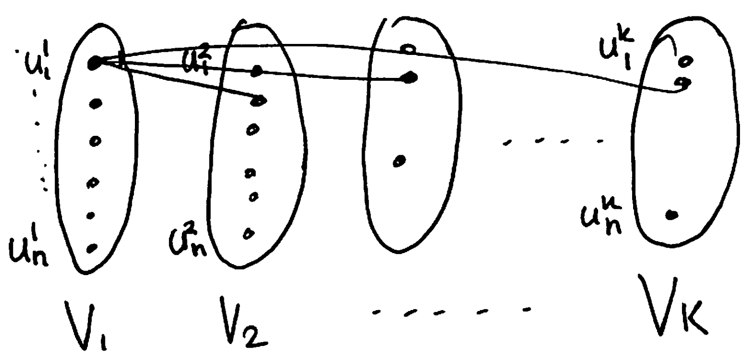
so given $\underbrace{(G, \{V_i\}_{i=1}^{k}, k)}_{MCC} \longmapsto \underbrace{(G, k)}_{C}$

1. $g(k) = k$ ✓
2. Yes by arg. above ✓
3. poly-time: linear ✓

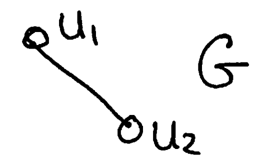
✓ MCC $\mapsto$ C

now, C $\longmapsto$ MCC? Given (G, k) need to produce $(G', \{V_i\}_{i=1}^{k}, k')$

$\tilde{G}$:



- need clique to use 1 vertex per set
- to represent all possible cliques in G need every vertex to be selectable from any set.



$V(G) = \{u_1, \dots, u_n\}$

add (u_ℓ^i, u_h^j) for $i \neq j$ and $(u_\ell, u_h) \in E(G)$

G has k-clique $\Rightarrow$ $\tilde{G}$ has a MC k-clique. Pick up j^{th} copy of the j^{th} vertex in clique ✓

$\tilde{G}$ has a MC-k-clique $\Rightarrow$ G has a k-clique. this works b/c u_j^i is not adj to u_j^ℓ for any ℓ.

In-Class Exercise

Prove that k-CLIQUE on regular graphs is at least as hard as kCLIQUE.

Need to map (G, k) to (G', k') so that G' is <u>degree-regular.</u>

Want any k-clique in G' to only use vertices / edges that correspond to those in G.

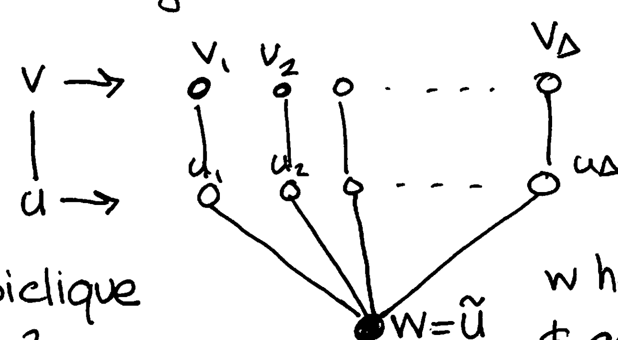
<u>Q1</u> What degree should G' have? Δ = max deg in G. $\leq n-1$

↑ don't have to delete edges ◡ (could choose something bigger)

need to start w/ G & add edges so all vertices get $\Delta - \deg(v)$ new edges <u>AND</u> any k-clique in G' <u>uses no new edges.</u>

suggestion: add vertices of deg 1 ← NOT regular.

<u>Trick</u> make Δ copies of G



k-clique in $G \Rightarrow$ one in each copy ✓

can a k-clique in $\hat{G}$ use a $\tilde{v}$ vertex? NO (unless $k=2$) <u>OK</u>

w has deg Δ & added 1 to $\deg(u_i)$ $\forall i$

If v has deg $d \Rightarrow$ add $\Delta - d$ new vertices $\tilde{v}_1, \ldots \tilde{v}_{\Delta-d}$ & create a biclique between $\{v_1, \ldots, v_\Delta\}$ and $\{\tilde{v}_1, \ldots, \tilde{v}_{\Delta-d}\}$. $\forall v \in G$

Replacing $P \neq NP$

NP = "non-deterministic polynomial" on Turing machines

- $P \longleftrightarrow FPT$

Definition: (short TM acceptance) Given (M, x, k),

input

parameter $\in \mathbb{N}$

(non-deterministic) TM

does M accept x (on some branch) in $\leq k$ steps?

(# states is bounded by $f(n)$ [$n = |\text{input}|$] $\Rightarrow$ in k steps, up to $f(n)^k$

XP

Theorists' Hypothesis: Short TM acceptance is not in FPT.

Engineers' Hypothesis: k-Clique is not in FPT

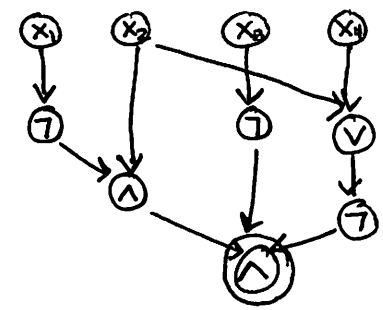
Thm: Theorists' Hyp. $\Longleftrightarrow$ Engineers' Hyp

Weighted Circuit SAT

Defn: a boolean circuit is a DAG w/ 3 types of nodes:

- nodes of indegree 0 (input)
- nodes of indegree 1 (negation)
- nodes of indegree >1 (and / or)

+ exactly one node of outdegree zero is labelled output



Problem: CIRCUIT SAT: Given a boolean circuit, does it have a satisfying assignment?

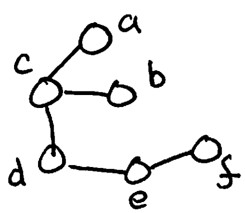
Problem: WEIGHTED CIRCUIT SAT: Is there a satisfying assignment of weight $\leq k$

↑ # TRUES.

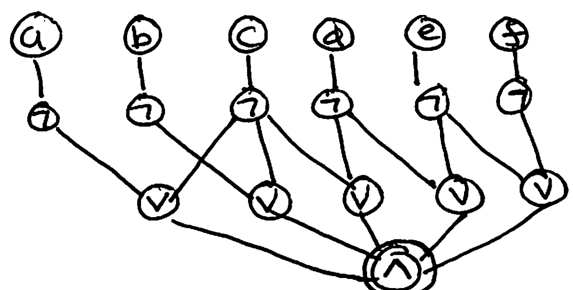
Example Reductions:

$(G, k) \rightarrow (C, k')$

vertices → inputs
(wt. k ⇒ set of k)

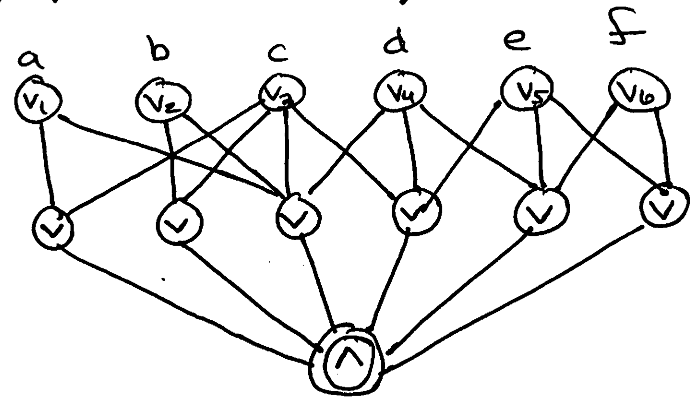


edge (u,v) ⇒ not u or not v



Independent Set

$(G, k) \longrightarrow (C, k)$



Dominating Set

Fine-grained Complexity

- The circuit for DOMSET was more "complex" than the one for INDSET, but how do we make this precise?

Defn The depth of a circuit is longest path from input to output

The weft of a circuit is max # of $\underbrace{\text{large}}_{>2 \text{ inputs (indegree)}}$ gates on a path from input to output

Examples: IS had depth 3, weft 1 ; DS had depth 2, weft 2

Defn the class $C[t,d]$ is the set of all boolean circuits of depth d, weft t

weft

cannot grow unbounded w/ instance size

The W Hierarchy P is the class $W[t]$ if $\exists d$ s.t. P is (param.) reducible to WCSAT on $C[t,d]$.

all circuits

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \dots \subseteq W[t] \subseteq W[P] \subseteq XP$$

k-IS/Clique

k-DomSet (on general graphs)

Problems

1. Show K-DOMINATINGSET and K-HITTINGSET are equally hard.

   given $\mathcal{F} = \{S_1, \ldots, S_n\}$ $\exists$ k elements $x_1 \ldots x_k$ st. $\forall i$, $S_i \ni x_j$ for some j.

2. Prove K-MULTICOLORED-GRID is W[1]-hard.

Defn Given a graph G and a k^2-vertex coloring $\{V_{i,j}\}_{1 \le i,j \le k}$, find a $k \times k$ grid as a subgraph of G so that vertex $v_{i,j}$ (row i, column j) is in $V_{i,j}$.

3. Prove PARTIALVERTEXCOVER parameterized by k is W[1]-hard by reduction from K-INDEPENDENTSET on regular graphs.

Defn Given a graph G and integers k, s, is there a ~~indepe~~ set of k vertices that covers at least s edges?

Note: you get to pick s in the reduction!

Dominating Set

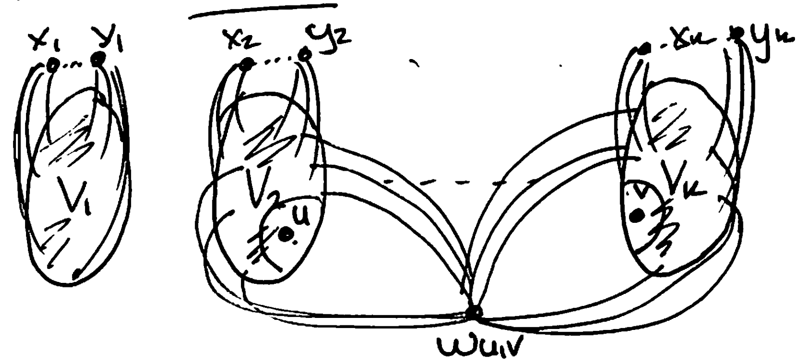
BONUS MATERIAL

Let's reduce from MULTICOLORED INDEPENDENT SET.

- Given (G, k) and $\{V_i\}_{i=1}^{k}$ a k-coloring $\Rightarrow$ produce (G', k') so G' has a k'-Domset $\Longleftrightarrow$ G had an indep set w/ 1 vertex in each V_i.

⓪ We'll let $k = k'$.

① Let's turn the V_i's into cliques (so we can pick one from each to dominate):



② We'll transform edges in G to vertices in G' so they can be dominated only if you didn't put both end points in your dom set. $(u,v) \rightarrow w_{uv}$ with $u \in V_i$, $v \in V_j$
w_{uv} is adj to all of $V_i \setminus \{u\}$ and $V_j \setminus \{v\}$. (so if you pick $u \in V_i$ and $v \in V_j$, $w_{u,v}$ is not dominated).

③ to make sure vertices $w_{u,v}$ aren't picked as dominators, we add two vertices x_i, y_i to G' for each i s.t. x_i is not a nbr. of y_i but x_i, y_i are adj to everything in V_i.
Then these aren't dominated by any w (or each other) $\Rightarrow$ must pick a vertex in each V_i.