Parameterized algorithms

and

Fine-grained complexity

[[Part 3: Methodology for lower bounds]]

Michał Pilipczuk

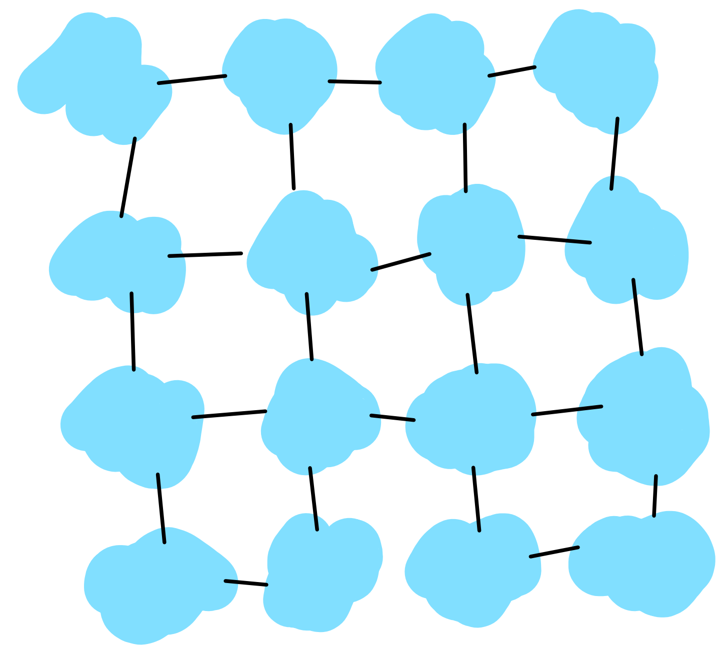
University of Warsaw

EPIT '23, Île d'Oléron

June 16^{th}, 2023

Addendum

Recall: G planar, $tw(G) > 9/2\, t \Rightarrow t \times t$ grid minor



Thm G planar, n vertices

$\Rightarrow tw(G) \leq O(\sqrt{n})$

Pf $tw(G) > 9/2 \sqrt{n+1}$

$\Rightarrow \sqrt{n+1} \times \sqrt{n+1}$ grid minor

$\Rightarrow \geq n+1$ vertices $\square$

Cor Vertex Cover, IndSet, HamCycle, DomSet, 3Coloring, ...
in time $2^{O(\sqrt{n})}$ on planar graphs.

Cor [Lipton-Tarjan] A planar graph has a balanced separator of size $O(\sqrt{n})$

Hardness for PA

	Problems	Efficient	Reductions	Assumption
Classic	$L \subseteq \{0,1\}^*$	ptime	ptime reductions	$SAT \notin P$
Parameterized	$P \subseteq \{0,1\}^* \times \mathbb{N}$	fpt (runtime $f(k) \cdot n^c$)	fpt reductions	$\text{Clique}/_k \notin FPT$ (no runtime $f(k) \cdot n^c$)

ptime red: $\left[K \xrightarrow{\text{ptime}} L ,\ L \in P \right] \Rightarrow K \in P$

fpt red: $\left[P \xrightarrow{\text{fpt}} Q ,\ Q \in FPT \right] \Rightarrow P \in FPT$

fpt reductions

Def $P \xrightarrow{\text{fpt}} Q$ if there is an algorithm $\mathcal{A}$ s.t.

$$(x,k) \xrightarrow{\mathcal{A}} (y,\ell)$$

i) $(x,k) \in P \iff (y,\ell) \in Q$

ii) runtime $g(k) \cdot |x|^d$ (g computable)

iii) $\ell \leq h(k)$ (h computable)

Lemma $[P \xrightarrow{\text{fpt}} Q,\ Q \in FPT] \Rightarrow P \in FPT$

Pf Q solvable in time $f(\ell) \cdot |y|^c \Rightarrow$

P solvable in time $f(h(k)) \cdot (g(k) \cdot |x|^d)^c = f'(k) \cdot |x|^{cd}$ □

Cor If P is hard for FPT, so is Q.

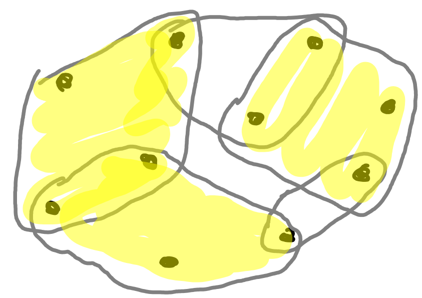
Note $P \xrightarrow{\text{fpt}} Q \xrightarrow{\text{fpt}} R \Rightarrow P \xrightarrow{\text{fpt}} R$

W[1]: lazy approach

Def $W[1] := [\text{Clique}/k]^{fpt} := \{ \text{all } P \text{ s.t. } P \xrightarrow{fpt} \text{Clique}/k \}$

Def P is W[1]-hard if $\text{Clique}/k \xrightarrow{fpt} P$

/* implying $W[1] \xrightarrow{fpt} P$ */



Now $\text{Clique}/k \xrightarrow{fpt} \text{Set Cover}/k$

Need!

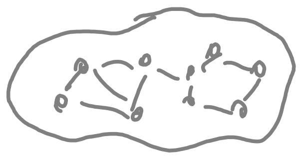
Step 1 $\text{Clique}/k \xrightarrow{fpt} \text{Ind Set}/k$

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

Hardness of Set Cover

$\text{IndSet}/k \xrightarrow{\text{fpt}} \text{Set Cover}/k$

Given: $G=(V,E)$, $k \in \mathbb{N}$

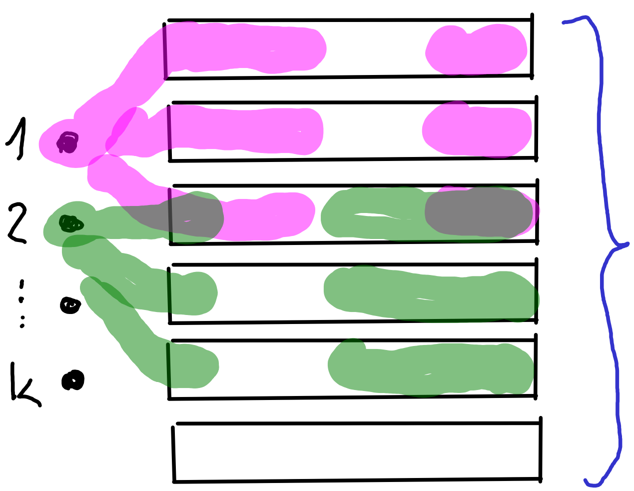


/* wlog no isolated vrts */

Construct: $U = \{1,2,\dots,k\} \uplus \{(e,\{i,j\}) : e \in E,\ 1 \le i < j \le k\}$

Add to $\mathcal{F}$ all sets of the form:

$X_{i,u} := \{i\} \cup \{\text{all } (e,\{i,j\}) \text{ s.t. } j \neq i \text{ and } e \text{ is } \underline{\underline{\text{not}}} \text{ incident to } u\}$



$\binom{k}{2}$ copies of E

Obs In solution of size k, need $X_{i,u}$ for $i=1,\dots,k$

Obs $X_{i,u} \cup X_{j,v}$ cover $\{(\cdot,\{i,j\})\}$

$\iff$ u,v are not adjacent.

Hardness of Set Cover

We proved Clique/k $\xrightarrow{\text{fpt}}$ Set Cover/k, so Set Cover/k is W[1]-hard

$\hookrightarrow$ No fpt algorithm unless FPT = W[1]

Principle

reductions from 3SAT $\rightsquigarrow$ encode search space $\{0,1\}^n$

reductions from Clique/k $\rightsquigarrow$ encode search space $\{1,\dots,n\}^k$

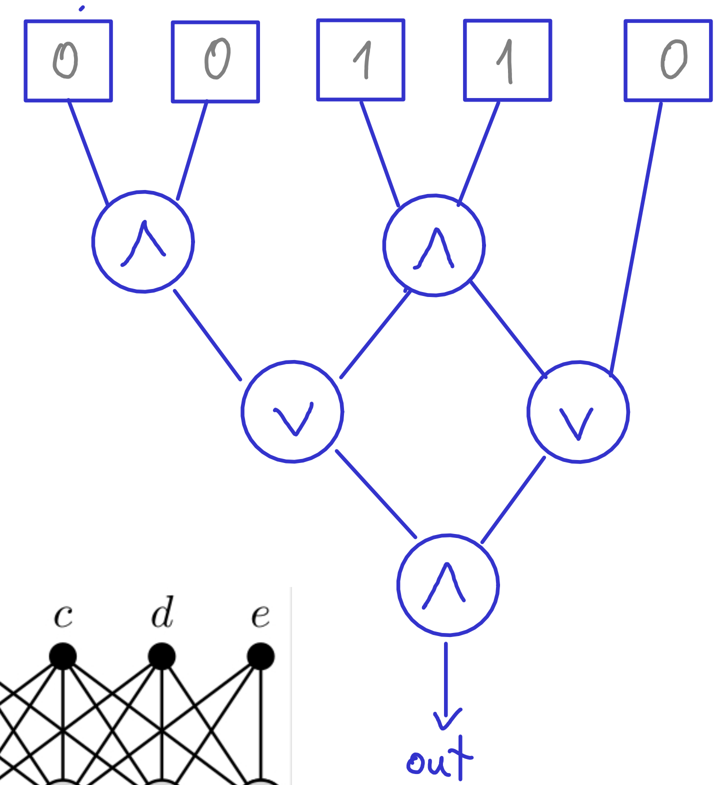
Q Does Set Cover/k belong to W[1]?

- We actually do not know.
- Set Cover/k is complete for W[2]

Lazy: $W[2] := [\text{Set Cover}/k]^{\text{fpt}}$

Now: Understand the difference between Clique/k and Set Cover/k

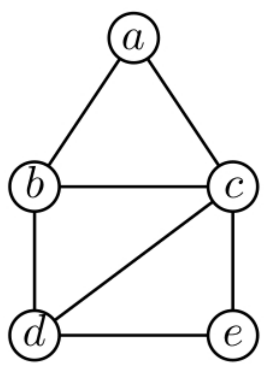
Circuits



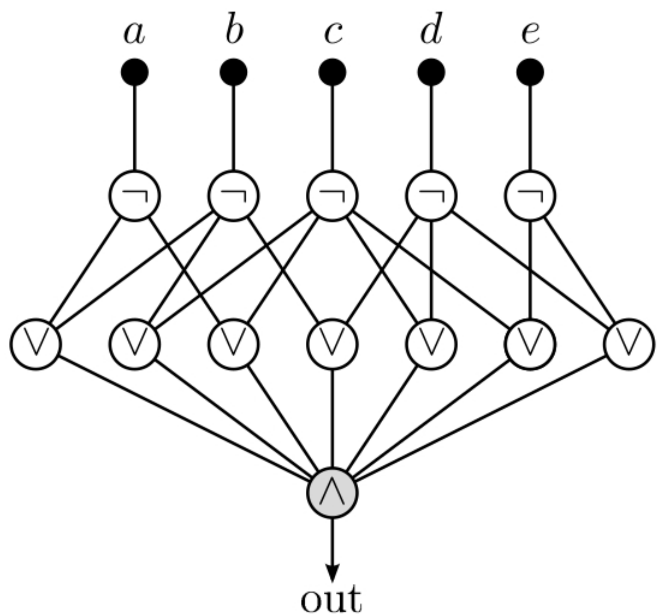
Weighted Circuit SAT:

Input: Circuit C, $k \in \mathbb{N}$

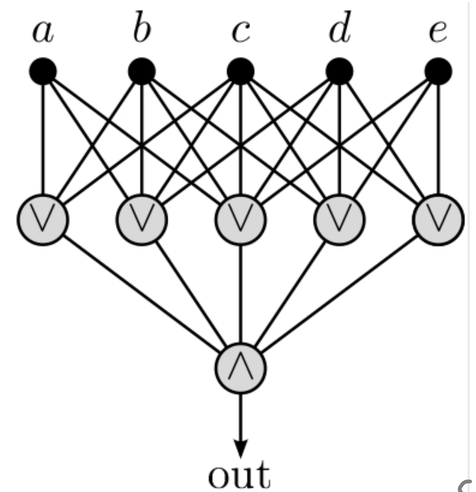
Question: Is there satisfying assignment with exactly k ones?



G



IndSet on G



Dominating Set on G

Source: Parameterized Algorithms

DomSet = set X s.t. $\bigcup_{u \in X} N[u] = V(G)$ = Set cover in $(V(G), \{N[u] : u \in V(G)\})$

Def Weft of C = max #unbounded fan-in gates on an input-output path

W-hierarchy

<u>Def</u> WCS[t,d] := Weighted Circuit Satisfiability for weft t and depth d

<u>Def</u> $W[t] := \bigcup_{d \in \mathbb{N}} [\, WCS[t,d] \,]^{fpt}$

$=$ problems fpt-reducible to WCS[t,d] for some $d \in \mathbb{N}$

<u>Thm</u>
- Clique/k, IndSet/k are W[1]-complete
- Set Cover/k, DomSet/k are W[2]-complete

$$\mathrm{FPT} \subseteq \underbrace{W[1] \subseteq W[2] \subseteq W[3] \subseteq \dots \subseteq \overset{\text{WCS for formulas}}{W[SAT]} \subseteq \overset{\text{WCS for circuits}}{W[P]}}_{\text{like NP}} \subseteq \underbrace{\overset{[\text{FO-MC}]^{fpt}}{AW[*]}}_{\text{like PSPACE} = \text{AP}} \subseteq XP$$

<u>Practice</u>:
- thousands of W[1]-hard problems
- hundreds of W[1]-complete problems
- tens of W[2]-complete problems
- hardly any W[t]-complete problems for $t \geq 3$

Fine-grained complexity

FVS/k : $(5k)^k \cdot n^{O(1)} \rightsquigarrow 2.7^k \cdot n^{O(1)} \overset{?}{\rightsquigarrow} 2^k \cdot n^{O(1)} \overset{???}{\rightsquigarrow} 2^{O(\sqrt{k})} \cdot n^{O(1)}$

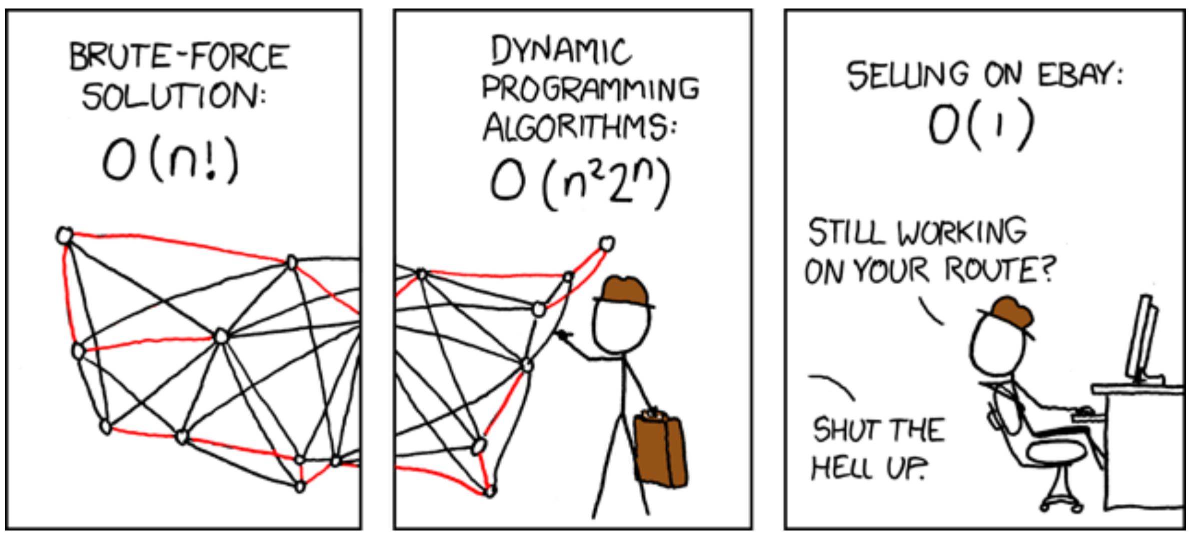
No $n^{O(1)}$

Clique/k : $n^{k+O(1)} \overset{?}{\rightsquigarrow} f(k) \cdot n^{O(\sqrt{k})}$

No $f(k) \cdot n^{O(1)}$

Need: Stronger Assumptions

Travelling Salesman:



source: xkcd 399

What about $2^{O(\sqrt{n})}$?

P vs NP : $n^{O(1)}$ vs no $n^{O(1)}$

FPT vs W[1] : $f(k) \cdot n^{O(1)}$ vs no $f(k) \cdot n^{O(1)}$

Exponential Time Hypothesis

Consider 3SAT instance φ with n variables and m clauses.

Brute force: $2^n \cdot \|\varphi\|^{O(1)}$

Simple: $2^{0.94n} \cdot \|\varphi\|^{O(1)} = 1.92^n \cdot \|\varphi\|^{O(1)}$

- as long as there is a clause $C = (x \vee \neg y \vee z)$, branch on 7 possibilities for x, y, z
- runs in time $2^{(\log_2 7)/3 \cdot n} \cdot \|\varphi\|^{O(1)} \leq 1.92^n \cdot \|\varphi\|^{O(1)}$

[PPSZ]: $2^{0.387n} \cdot \|\varphi\|^{O(1)} = 1.31^n \cdot \|\varphi\|^{O(1)}$

Also $(2-\varepsilon_k)^n \cdot \|\varphi\|^{O(1)}$ for k-SAT, but ε_k gets smaller with k

Def $\delta_k := \inf\{c : \text{there is a } 2^{cn} \cdot \|\varphi\|^{O(1)} \text{ for k-SAT}\}$

ETH: $\delta_3 > 0$

$\exists_{\delta>0}$ 3SAT cannot be solved in time $O(2^{\delta n})$

SETH: $\lim_{k\to\infty} \delta_k = 1$

Savings for k-SAT tend to 0

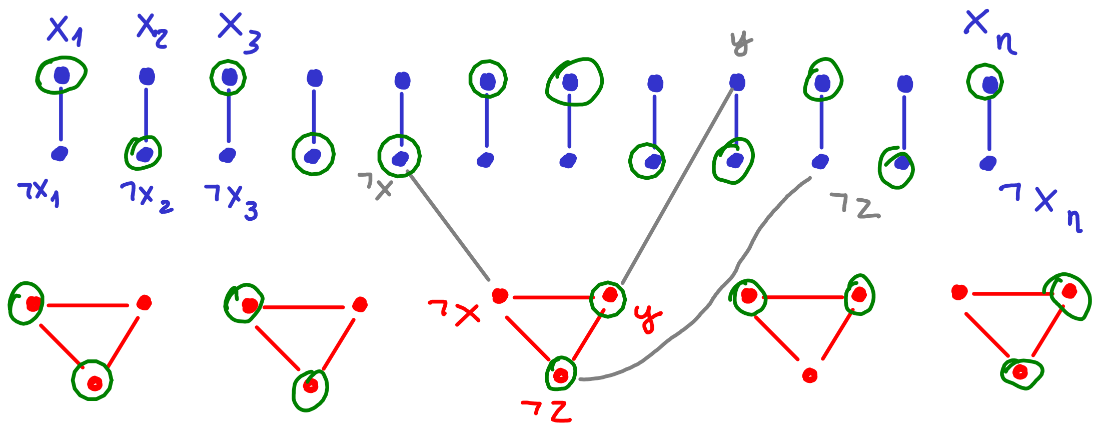
Note: ETH $\Rightarrow$ 3SAT not in time $2^{o(n)}$

Note: SETH $\Rightarrow$ SAT not in time $(2-\varepsilon)^n \cdot \|\varphi\|^{O(1)}$ for any $\varepsilon > 0$.

Hardness for Vertex Cover

3SAT $\longrightarrow$ Vertex Cover

φ $\qquad\qquad$ G_φ



φ satisfiable $\iff$ G_φ has a vertex cover of size $k = n + 2m$

G_φ has $N = 2n + 3m$ vrts

$2^{o(N)}$-time for VC $\Rightarrow$ $2^{o(n+m)}$ for 3SAT

Issue: We only assumed $2^{o(n)}$ hardness for 3SAT

Need: Hardness expressed in formula size.

Sparsification Lemma

Thm [Impagliazzo, Paturi, Zane]

Fix $k, \varepsilon > 0$.

Given kCNF formula φ, one can compute $\psi_1, \psi_2, \dots, \psi_t$ on same vars s.t.

a) $t \le 2^{\varepsilon n}$ and in time $O(2^{\varepsilon n})$

b) φ satisfiable $\Longleftrightarrow$ one of $\psi_1, \dots, \psi_t$ satisfiable

c) In each ψ_i, every variable appears $\le C(k,\varepsilon)$ times. $\Longrightarrow$ # clauses $\le Cn$

Pf Branch on variables appearing too often $\Rightarrow$ Progress $\square$

Corollary ETH $\Rightarrow$ $\exists_{\delta>0}$ 3SAT cannot be solved in time $O(2^{\delta(n+m)})$.

Pf Apply Sparsification Lemma, solve each ψ_i using $\cdot$. $\square$

Corollary Under ETH, no $2^{o(N+M)}$ algorithm for

Vertex Cover, IndSet, Clique, HamCycle, TSP, 3-Coloring, DomSet, FVS, ...

Note: also no $2^{o(k)} \cdot N^{O(1)}$, would be sufficient to have $k = O(n+m)$

$N = (n+m)^{O(1)}$

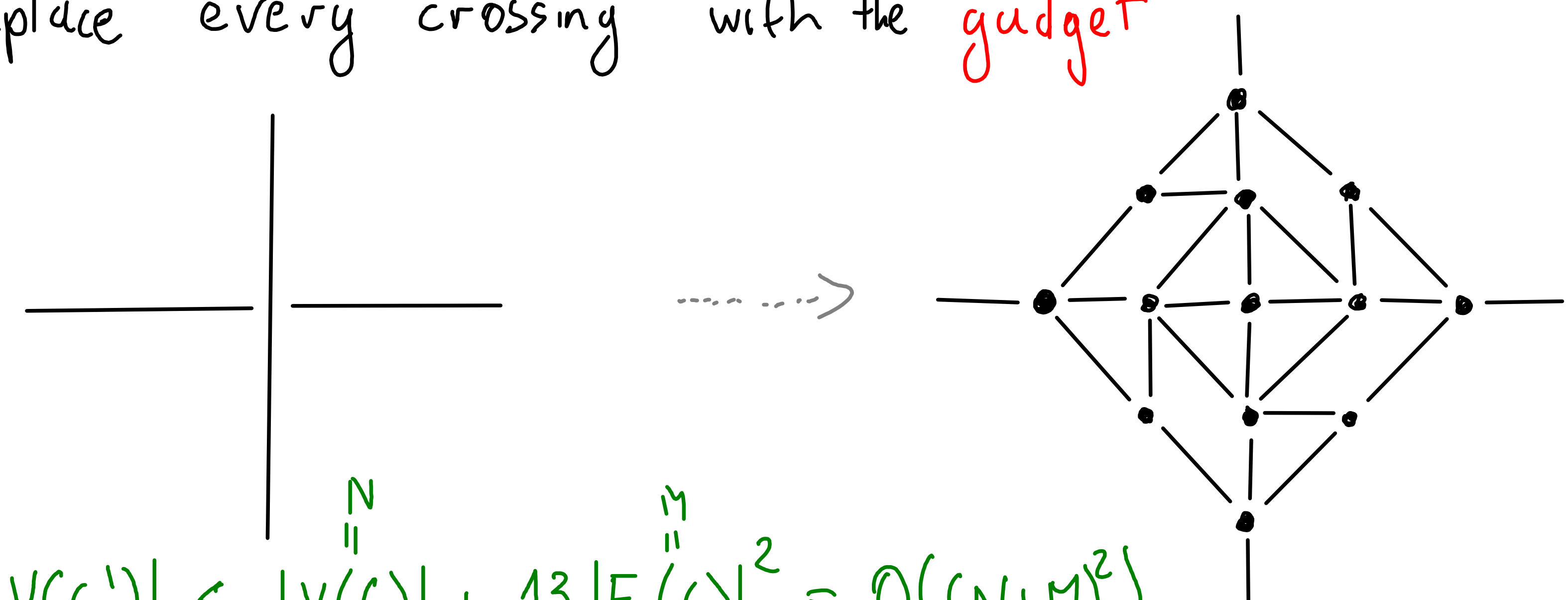
Planar problems

Recall Planar 3-Coloring can be solved in time $2^{O(\sqrt{N})}$.

$$\underset{G}{\text{3-Coloring}} \longrightarrow \underset{G'}{\text{Planar 3-Coloring}}$$

i) Draw G in $\mathbb{R}^2$ with crossings

ii) Replace every crossing with the gadget



Note: $|V(G')| \leq \underset{N}{\underset{\|}{|V(G)|}} + 13\,\underset{M}{\underset{\|}{|E(G)|}}^2 = O((N+M)^2)$.

Cor $2^{o(\sqrt{N})}$ for Planar 3Col $\Rightarrow$ $2^{o(N+M)}$ for 3Col $\Rightarrow$ $\neg$ETH.

$\hookrightarrow$ Same for other planar problems.

ETH and W[1]

Thm [Chen et al]

Assuming ETH, no $f(k) \cdot n^{o(k)}$-time for Clique, for any computable f.

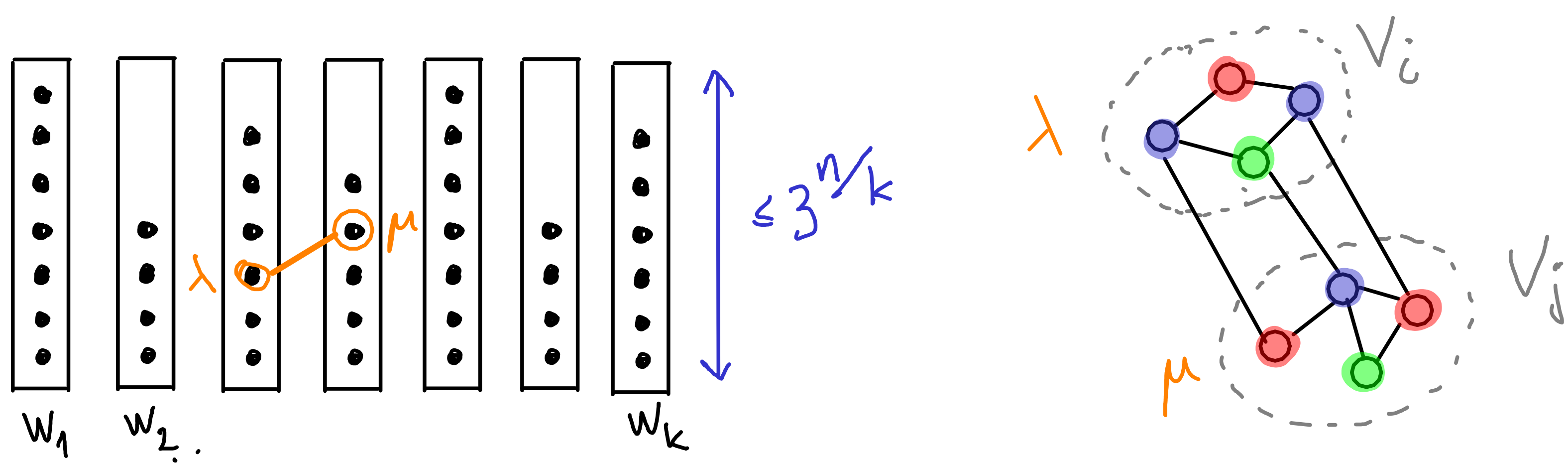
Pf Assume $f(k) = 2^{2^k}$. Take instance G of 3Coloring.

i) $k := \frac{1}{2} \lg n$.

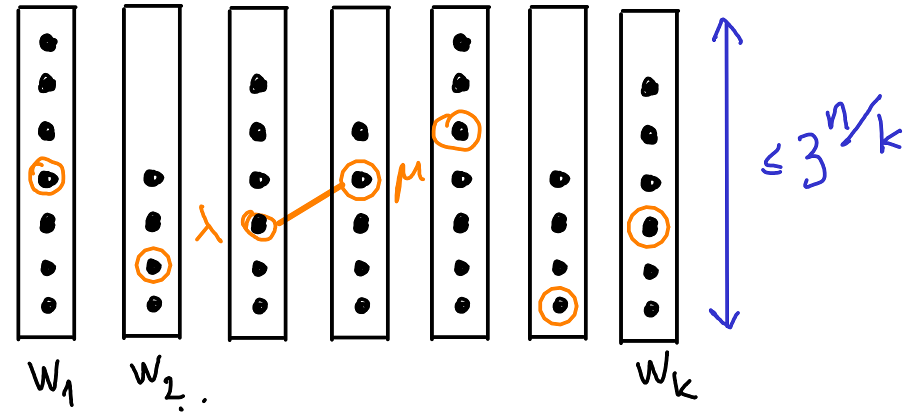
ii) Partition $V(G)$ into $V_1, V_2, \ldots, V_k$, each of size n/k.

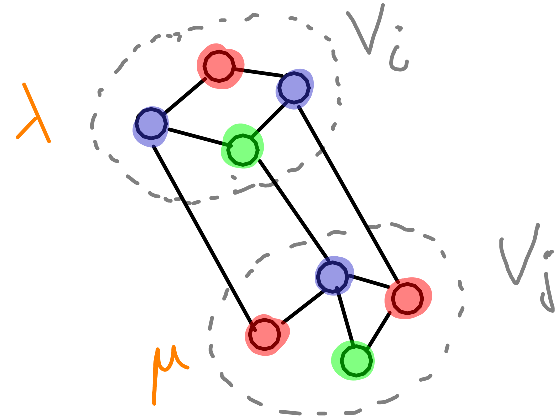
iii) For each V_i, enumerate all $\leq 3^{n/k}$ 3-colorings of $G[V_i]$. $\rightsquigarrow W_i$

iv) Connect $\lambda \in V_i$ with $\mu \in V_j$ iff $\lambda \cup \mu$ consistent in $G[V_i \cup V_j]$



ETH and W[1]





$H :=$ obtained graph

Claim G has a 3-coloring $\iff$ H has a k-clique.

Pf $\Rightarrow$ Restrict 3-coloring to V_i-s

$\Leftarrow$ Take union of colorings in V_i-s $\square$

$$N := |V(H)| = k \cdot 3^{n/k}$$

$k = \frac{1}{2} \lg n$

Runtime for 3Col ?

$$f(k) \cdot N^{o(k)} = 2^{2^k} \cdot \left(k \cdot 3^{n/k}\right)^{o(k)} = 2^{\sqrt{n}} \cdot \left(\tfrac{1}{2} \lg n\right)^{o(\lg n)} \cdot 2^{n \cdot \frac{o(\lg n)}{\lg n}} = 2^{o(n)}$$

If $f(k) = 2^{2^{2^k}}$, then $k = \frac{1}{2} \lg \lg n$, and so on... $\square$

ETH in general

Usage: Asymptotics of the exponents.

Quality of lower bound $\leftrightsquigarrow$ Parameter blow-up in reduction

$k = \Theta(n+m) \Rightarrow$ No $2^{o(k)} \cdot N^{O(1)}$-time algo

$k = \Theta((n+m)^2) \Rightarrow$ No $2^{o(\sqrt{k})} \cdot N^{O(1)}$-time algo

Slighty superexponential: $k^{O(k)} \cdot n^{O(1)} = 2^{O(k \lg k)} \cdot n^{O(1)}$

- Example: Delete k vertices to get maxdeg $< k$.
- Simple $k^k \cdot N^{O(1)}$ branching
- No $2^{o(k \lg k)} \cdot N^{O(1)}$ under ETH (Lokshtanov et al.) reduction: $k = O(n/\lg n)$

Doubly exponential Example: Edge Clique Cover

- $2^{2^k} \cdot N^{O(1)}$-time algo (Gramm et al.)
- No $2^{2^{o(k)}} \cdot N^{O(1)}$ under ETH (Cygan et al.) reduction: $k = O(\lg n)$

Strong ETH

Recall: $\delta_k = \inf\{c : k\text{-SAT in time } O(2^{cn})\}$

SETH $\iff \lim \delta_k = 1 \implies$ CNF-SAT not in $O((2-\varepsilon)^n)$

Usage: Exact bases of exponents

Issue: Need to control parameter blow-up exactly

Examples:

- concrete problems parameterized by treewidth (Lokshtanov et al.)
  - Ind Set: no $(2-\varepsilon)^t \cdot n^{O(1)}$
  - q-Coloring: no $(q-\varepsilon)^t \cdot n^{O(1)}$
  - Ham Cycle: no $(2+\sqrt{2}-\varepsilon)^t \cdot n^{O(1)}$, $(2+\sqrt{2})^{pw} \cdot n^{O(1)}$ known
- Set Cover/$|\mathcal{F}|$: no $(2-\varepsilon)^{|\mathcal{F}|} \cdot (|U|+|\mathcal{F}|)^{O(1)}$
- Set Cover/$|U|$: Set Cover Conjecture (Cygan et al.)
  - $\hookrightarrow$ no $(2-\varepsilon)^{|U|} \cdot (|U|+|\mathcal{F}|)^{O(1)}$

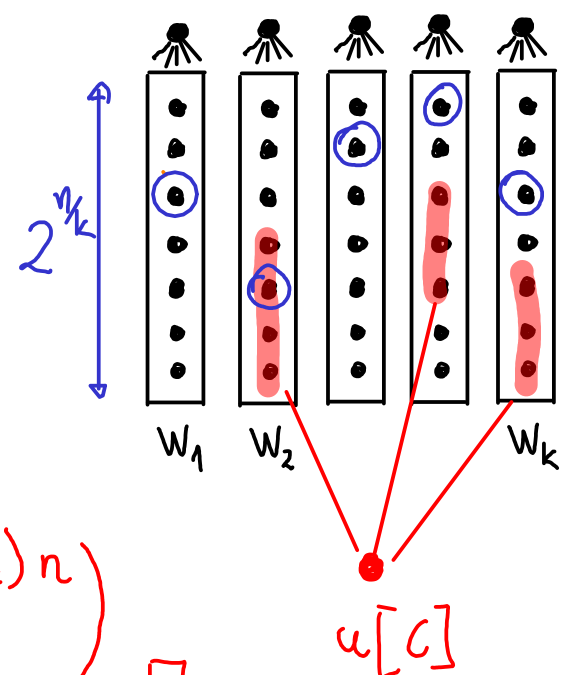
equivalent mod 2

SETH $\Rightarrow$ ptime lower bounds

Thm [Pătraşcu, Williams]

Let $k \geq 3$. Under SETH, no $O(N^{k-\varepsilon})$-time for k-DomSet.

Pf
- Start with CNF-SAT instance φ with $|\text{Vars}| = n$
- Divide Vars into $V_1, \dots, V_k$ of size n/k each.
- For every i, generate all $2^{n/k}$ assignments to V_i. $\rightsquigarrow W_i$
- Make W_i into a clique, add universal vrt
- For clause C, construct $u[C]$ adjacent to all partial assignments satisfying C



Obs φ satisfiable $\iff$ $\exists$ domset of size k

Runtime $O(N^{k-\varepsilon}) = O((2^{n/k})^{k-\varepsilon}) = O(2^{(1-\varepsilon/k)n})$ $\square$

Fine-grained Clique

Could we prove a similar lower bound for Clique?

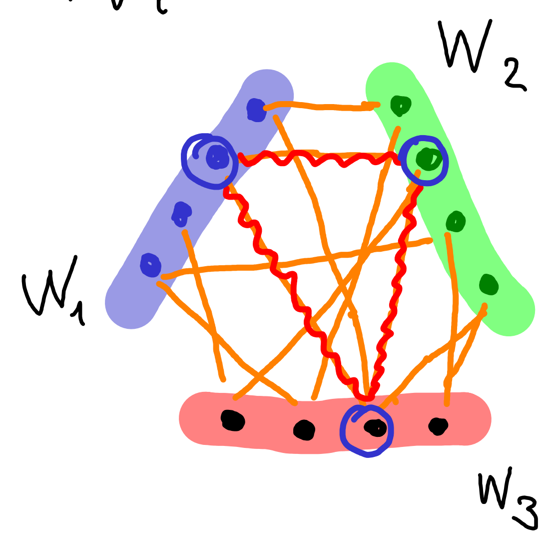
Thm [Nešetřil, Poljak]
Clique can be solved in time $O\left(N^{\frac{\omega}{3}\cdot k + O(1)}\right) = O\left(N^{0.77k}\right)$.

Pf – Divide $V(G)$ into V_1, V_2, V_3 of sizes $N/3$ at random.

↳ With prob. $\geq 1/\text{poly}(k)$, solution split $k/3, k/3, k/3$

– Enumerate all $k/3$ cliques in each V_i. ⇝ W_i

– Connect fully adjacent cliques in diff W_i's



Obs G has a k-clique ⇕ H has a triangle. → In time $O\left(\left(N^{k/3}\right)^{\omega}\right)$ by diagonal of $A(H)^3$. □

Orthogonal Vectors

$d = 5$

Input: Vectors $a_1, a_2, \ldots, a_n \in \{0,1\}^d$

$b_1, b_2, \ldots, b_n \in \{0,1\}^d$

$$a_i \begin{cases} 01101 \\ 11011 \\ 10001 \end{cases} \qquad b_i \begin{cases} 10110 \\ 01110 \\ 10101 \end{cases}$$

Question: Are there i, j s.t. $\langle a_i, b_j \rangle = 0$?

/* a_i, b_j do not simultaneously have 1s on same coordinate */

Trivial: Runtime $O(d \cdot n^2)$

OV Conjecture: $\forall_{\varepsilon > 0} \; \nexists_{c \in \mathbb{N}} \; O(n^{2-\varepsilon})$ algorithm for OV with $d = c \log n$.

Thm SETH $\Rightarrow$ OV Conjecture

SETH ⇒ OV Conjecture

φ: instance of CNF-SAT with n variables

- partition $\mathrm{Vars}(\varphi)$ into V_1 and V_2 of size $n/2$
- enumerate all $2^{n/2} = N$ variable assignments in V_1 and V_2 ⟶ W_1 and W_2
- for each $\alpha \in W_1$, make a vector $a_\alpha \in \{0,1\}^{\#\text{clauses}}$ indicating clauses <u>not</u> satisfied by α

$$
a_\alpha: \quad 2^{n/2}\left\{\begin{array}{c}
\overset{\#\text{clauses}}{\longleftrightarrow} \\
1011011 \\
0110110 \\
1101010 \\
0110110 \\
1110011
\end{array}\right.
\qquad\qquad
b_\alpha: \quad 2^{n/2}\left\{\begin{array}{c}
\overset{\#\text{clauses}}{\longleftrightarrow} \\
1100010 \\
1101111 \\
0101010 \\
1011111 \\
0010111
\end{array}\right.
$$

Dimension:

$m \leq c \cdot \log N$, because we can start with k-SAT and apply Sparsification Lemma

<u>Obs</u>: φ satisfiable $\Leftrightarrow$ OV instance has a solution.

Runtime $O(N^{2-\varepsilon}) = O\left(\left(2^{n/2}\right)^{2-\varepsilon}\right) = O\left(2^{(1-\varepsilon/2)n}\right)$ ↯

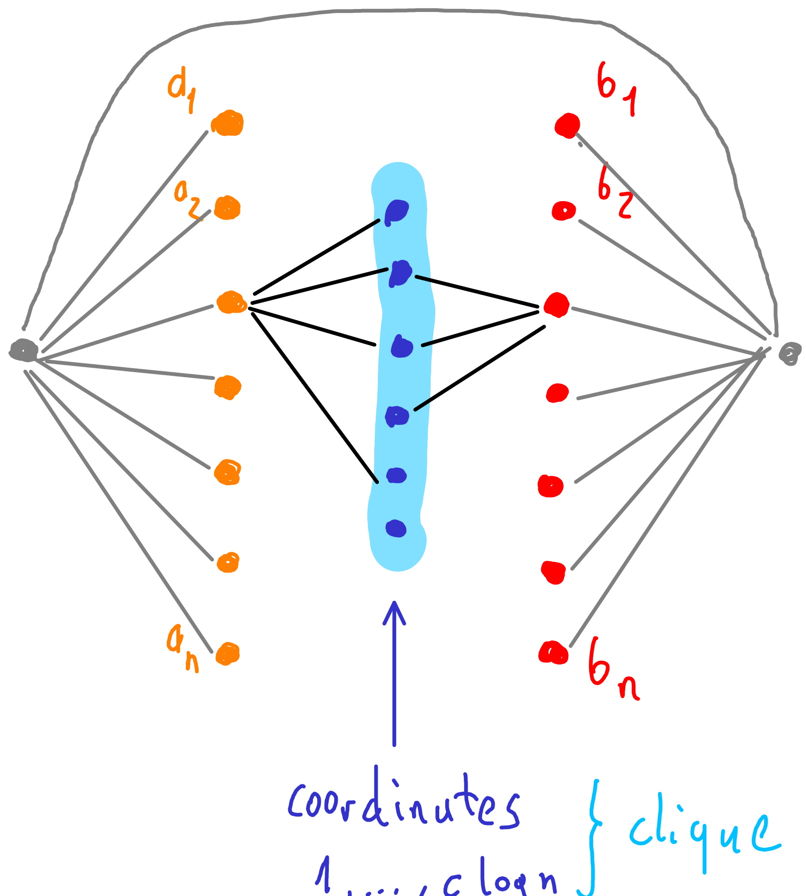
OV ⇒ Diameter

Thm Under OV Conjecture, distinguishing diam 2 from diam 3 cannot be done in time $O(n^{2-\varepsilon})$ even if $m = O(n)$.

Pf

Connect a_i to 1 entries in a_i

Connect b_j to 1 entries in b_j



Obs OV instance has a solution ⇔ Diameter = 3 ∎

Fine-grained complexity

Programme of obtaining tight upper and lower bounds on the complexity of problems.

Optimality programme in PC

Lower bounds within P

Assumptions:

- ETH, SETH, SeCoCo
- OV, Radius, Diameter not in subquadratic time
- 3Sum not in subquadratic time (given $L \subseteq \mathbb{Z}$, $\exists_{a,b,c \in L}\ a+b=c$?)
- APSP $\Leftrightarrow$ Negative Triangle $\Leftrightarrow$ Boolean Matrix Multiplication not in subcubic time (Vassilevska-Williams, Williams)

Example advances:

- Edit Distance not in time $O(n^{2-\varepsilon})$. [Backurs, Indyk]
- Diameter in $tw^{O(tw)} \cdot n^{1+o(1)}$, but not in $2^{o(tw)} \cdot n^{2-\varepsilon}$. [Husfeldt]
- Lower bounds for dynamic data structures.
- Work around (min,+)-convolution, Subset Sum, etc.

Conclusion

Parameterized Complexity:

- measure complexity in auxiliary parameters
- fpt : runtime $f(k) \cdot n^c$ xp: runtime $f(k) \cdot n^{g(k)}$
- multitude of techniques
- W-hardness : Methodology for lower bounds

Fine-grained Complexity

- Goal: Get tight bounds on complexity of problems
- Assumptions: ETH, SETH, ... $\rightsquigarrow$ Reductions
- May lead to algorithmic developments

Messages

- Complexity Theory subservient to Algorithm Design
- It's a jungle.

Thanks!