Fine-Grained Complexity

Édouard Bonnet

LIP, ENS Lyon

April 8th, 2025, Complexity Days, Lyon

Fine-grained complexity

Quantitative complexity theory:

- For some reference problems, assume that the current best algorithm cannot be significantly improved.
- Via reductions, derive consequences for the other problems Π.
- Forget about complexity classes.

Fine-grained complexity

Quantitative complexity theory:

- For some reference problems, assume that the current best algorithm cannot be significantly improved.
- Via reductions, derive consequences for the other problems Π.
- Forget about complexity classes.

Goal:

- Get an $f(n, \overline{k})$-time algorithm for Π.
- Show a $g(n, \overline{k})$-time algorithm for Π would improve the complexity of a reference problem.
- Make f and g as close as possible.

Immediate consequence of ETH for graph problems

ETH $= \exists s > 0$ s.t. n-variable 3-SAT is not in TIME(2^{sn})

n-variable 3-SAT instances may have $m = \Theta(n^3)$ clauses

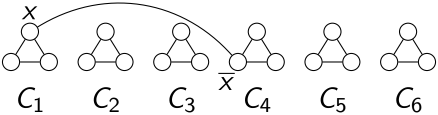
Immediate consequence of ETH for graph problems

ETH $= \exists s > 0$ s.t. n-variable 3-SAT is not in TIME(2^{sn})

n-variable 3-SAT instances may have $m = \Theta(n^3)$ clauses

n-variable m-clause $3\text{-SAT} \xrightarrow{\rho}_P O(n+m)$-vertex instance of Π

ρ only implies: ETH $\Rightarrow \Pi$ requires $2^{\Omega(n^{1/3})}$ time on n-vertex graph



$\Pi =$ Max Independent Set, built graphs have $3m$ vertices

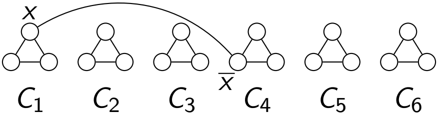
Immediate consequence of ETH for graph problems

ETH $= \exists s > 0$ s.t. n-variable $3\text{-}\mathrm{SAT}$ is not in TIME(2^{sn})

n-variable $3\text{-}\mathrm{SAT}$ instances may have $m = \Theta(n^3)$ clauses

n-variable m-clause $3\text{-}\mathrm{SAT} \xrightarrow{\rho}_P O(n+m)$-vertex instance of Π

ρ only implies: ETH $\Rightarrow \Pi$ requires $2^{\Omega(n^{1/3})}$ time on n-vertex graph



$\Pi =$ Max Independent Set, built graphs have $3m$ vertices

n-variable $3\text{-}\mathrm{SAT} \longrightarrow_P O(n)$-clause $3\text{-}\mathrm{SAT}$?

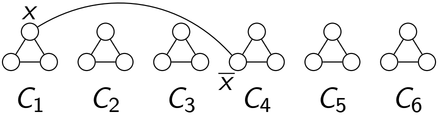
Immediate consequence of ETH for graph problems

ETH $= \exists s > 0$ s.t. n-variable 3-SAT is not in TIME(2^{sn})

n-variable 3-SAT instances may have $m = \Theta(n^3)$ clauses

n-variable m-clause $3\text{-SAT} \xrightarrow{\rho}_P O(n+m)$-vertex instance of Π

ρ only implies: ETH $\Rightarrow \Pi$ requires $2^{\Omega(n^{1/3})}$ time on n-vertex graph



$\Pi =$ Max Independent Set, built graphs have $3m$ vertices

n-variable $3\text{-SAT} \longrightarrow_P O(n)$-clause 3-SAT? Unlikely

Sparsifying k-SAT

Theorem (Impagliazzo–Paturi–Zane '01)

For every $\varepsilon > 0$, given an n-variable k-SAT formula φ, $t \leqslant 2^{\varepsilon n}$ k-SAT formulas $\varphi_1, \ldots, \varphi_t$ can be computed in time $n^{O(1)} 2^{\varepsilon n}$ s.t.

- *φ is satisfiable iff at least one φ_i is satisfiable, and*
- *for every $i \in [t]$, φ_i has at most at most $C_{\varepsilon,k} n$ clauses.*

Sparsifying k-SAT

Theorem (Impagliazzo–Paturi–Zane '01)

For every $\varepsilon > 0$*, given an* n*-variable* k-SAT *formula* φ*,* $t \leqslant 2^{\varepsilon n}$ k-SAT *formulas* $\varphi_1, \ldots, \varphi_t$ *can be computed in time* $n^{O(1)} 2^{\varepsilon n}$ *s.t.*

- φ *is satisfiable iff at least one* φ_i *is satisfiable, and*
- *for every* $i \in [t]$*,* φ_i *has at most at most* $C_{\varepsilon,k} n$ *clauses.*

Theorem

ETH $\Rightarrow$ m*-clause* 3-SAT *requires* $2^{\Omega(m)}$ *time.*

Sparsifying k-SAT

Theorem (Impagliazzo–Paturi–Zane '01)

For every $\varepsilon > 0$, given an n-variable k-SAT formula φ, $t \leqslant 2^{\varepsilon n}$ k-SAT formulas $\varphi_1, \ldots, \varphi_t$ can be computed in time $n^{O(1)} 2^{\varepsilon n}$ s.t.

- *φ is satisfiable iff at least one φ_i is satisfiable, and*
- *for every $i \in [t]$, φ_i has at most at most $C_{\varepsilon,k} n$ clauses.*

Theorem

ETH $\Rightarrow$ m-clause 3-SAT requires $2^{\Omega(m)}$ time.

Let $s > 0$ be such that n-variable 3-SAT is not in $\mathrm{TIME}(2^{sn})$
Set $\varepsilon := \frac{s}{3}$ and $\varepsilon' := \frac{s}{3C_{\varepsilon,3}}$

Sparsifying k-SAT

Theorem (Impagliazzo–Paturi–Zane '01)

For every $\varepsilon > 0$, given an n-variable k-SAT formula φ, $t \leqslant 2^{\varepsilon n}$ k-SAT formulas $\varphi_1, \ldots, \varphi_t$ can be computed in time $n^{O(1)} 2^{\varepsilon n}$ s.t.

- *φ is satisfiable iff at least one φ_i is satisfiable, and*
- *for every $i \in [t]$, φ_i has at most at most $C_{\varepsilon,k} n$ clauses.*

Theorem

ETH $\Rightarrow$ m-clause 3-SAT requires $2^{\Omega(m)}$ time.

Let $s > 0$ be such that n-variable 3-SAT is not in TIME(2^{sn})
Set $\varepsilon := \frac{s}{3}$ and $\varepsilon' := \frac{s}{3C_{\varepsilon,3}}$

Sparsification $+$ $2^{\varepsilon' m}$-time algorithm on each instance

$$n^{O(1)} 2^{\varepsilon n} + 2^{\varepsilon n} \cdot 2^{\varepsilon' C_{\varepsilon,3} n} < 2^{sn}$$

Tight ETH lower bounds

Theorem

ETH $\Rightarrow \Pi$ *requires* $2^{\Omega(n)}$ *time on n-vertex graphs for* $\Pi =$ Max Independent Set, Dominating Set, 3-Coloring, Vertex Cover, Feedback Vertex Set, Hamiltonian Cycle, *etc.*

Tight ETH lower bounds

Theorem

ETH $\Rightarrow \Pi$ requires $2^{\Omega(n)}$ time on n-vertex graphs for $\Pi =$ Max Independent Set, Dominating Set, 3-Coloring, Vertex Cover, Feedback Vertex Set, Hamiltonian Cycle, *etc.*

REDUCIBILITY AMONG COMBINATORIAL PROBLEMS†

Richard M. Karp

University of California at Berkeley

Abstract: A large class of computational problems involve the determination of properties of graphs, digraphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements of other countable domains. Through simple encodings from such domains into the set of words over a finite alphabet these problems can be converted into language recognition problems, and we can inquire into their computational complexity. It is reasonable to consider such a problem satisfactorily solved when an algorithm for its solution is found which terminates within a number of steps bounded by a polynomial in the length of the input. We show that a large number of classic unsolved problems of covering, matching, packing, routing, assignment and sequencing are equivalent, in the sense that either each of them possesses a polynomial-bounded algorithm or none of them does.

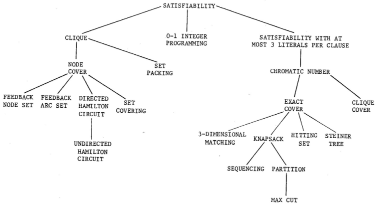


FIGURE 1 - Complete Problems

RICHARD M. KARP

Tight ETH lower bounds on planar graphs

All these problems admit $2^{O(\sqrt{n})}$-time algorithms in planar graphs

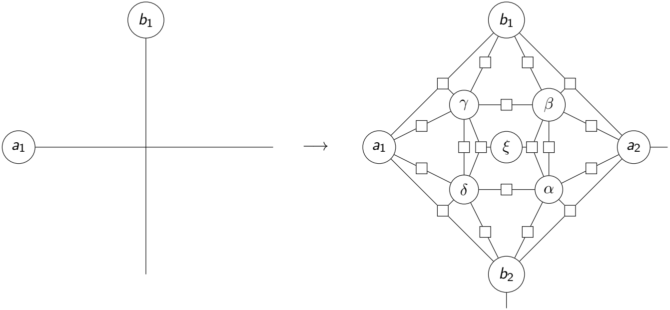
Tight ETH lower bounds on planar graphs

All these problems admit $2^{O(\sqrt{n})}$-time algorithms in planar graphs

$\textsc{Planar 3-SAT}$: the variable-clause incidence graph is planar

Theorem (Lichtenstein '82)

$\exists \rho$, *n-clause* $\text{3-SAT} \xrightarrow{\rho}_P$ $O(n^2)$*-variable* $\textsc{Planar 3-SAT}$.



9-variable 18-clause planar instance that forces $a_1 = a_2$, $b_1 = b_2$

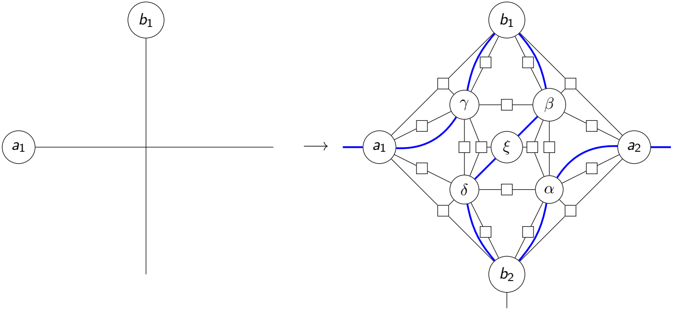
Tight ETH lower bounds on planar graphs

All these problems admit $2^{O(\sqrt{n})}$-time algorithms in planar graphs

$\textsc{Planar 3-SAT}$: the variable-clause incidence graph is planar

Theorem (Lichtenstein '82)

$\exists \rho$, *n-clause* $3\text{-SAT} \xrightarrow{\rho}_P O(n^2)$*-variable* $\textsc{Planar 3-SAT}$.



9-variable 18-clause planar instance that forces $a_1 = a_2$, $b_1 = b_2$

Tight ETH lower bounds on planar graphs

Theorem

ETH $\Rightarrow \Pi$ *requires* $2^{\Omega(\sqrt{n})}$ *time on n-vertex planar graphs for* $\Pi =$ Max Independent Set, Dominating Set, 3-Coloring, Vertex Cover, Feedback Vertex Set, Hamiltonian Cycle, *etc.*

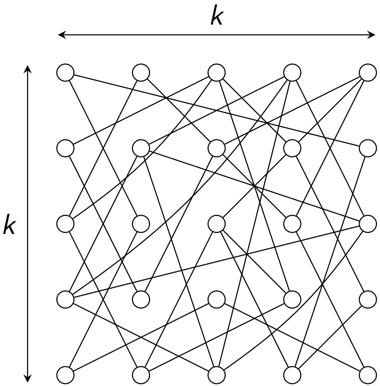
Can we explain other kinds of running times?

Theorem (Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk '11, Cut&Count)

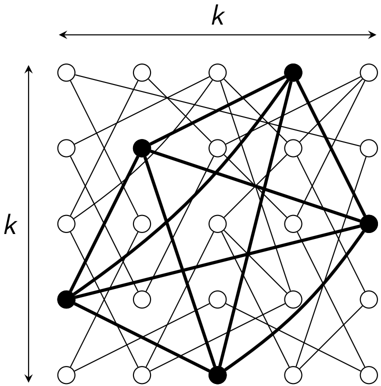
Hamiltonian Cycle, Feedback Vertex Set, Connected Dominating Set *can be solved in* $2^{O(w)}n^{O(1)}$ *on n-vertex graphs of treewidth* w.

For Chromatic Number, Disjoint Paths, Cycle Packing $2^{O(w \log w)}n^{O(1)}$ remains best.

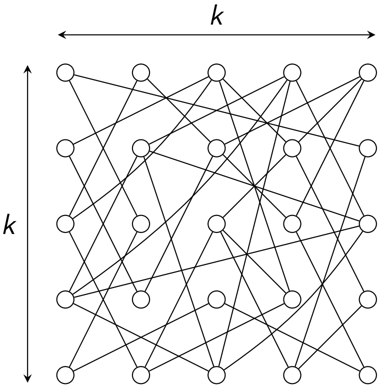
Permutation $k \times k$ Clique



Permutation $k \times k$ Clique



Permutation $k \times k$ Clique

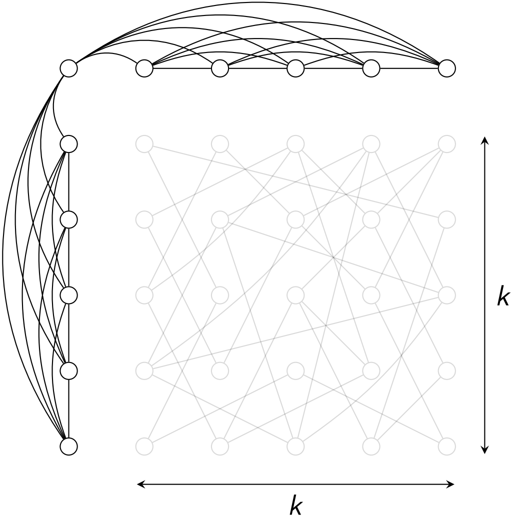


Theorem (Lokshtanov–Marx–Saurabh '11)

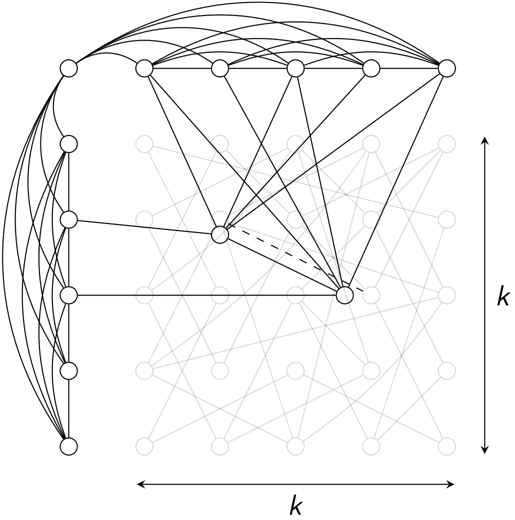
ETH $\Rightarrow$ Permutation $k \times k$ Clique *requires* $2^{\Omega(k \log k)}$ *time.*

Reduce from 3-Coloring with q batches of size $\frac{n}{q}$ s.t. $q \approx 3^{n/q}$.

CHROMATIC NUMBER parameterized by treewidth

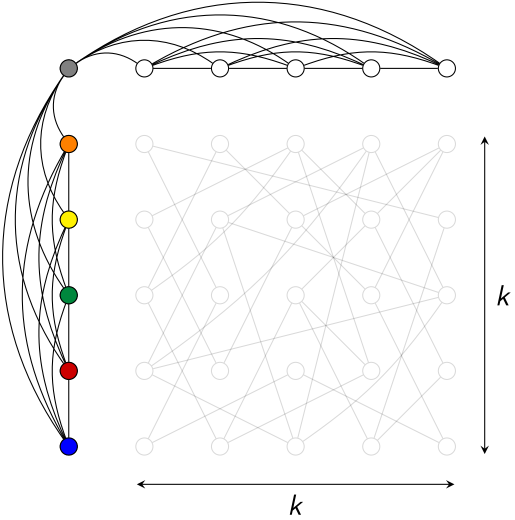


Chromatic Number parameterized by treewidth



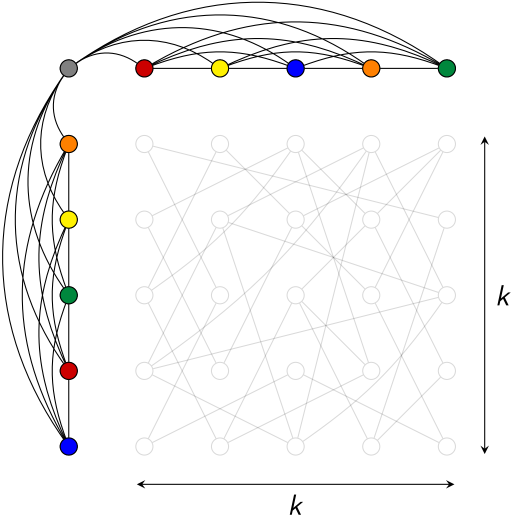
For every non-edge $(i,j)(i' \neq i, j' \neq j)$, add two adjacent vertices linked to their row vertex and to every column vertex but theirs.

CHROMATIC NUMBER parameterized by treewidth



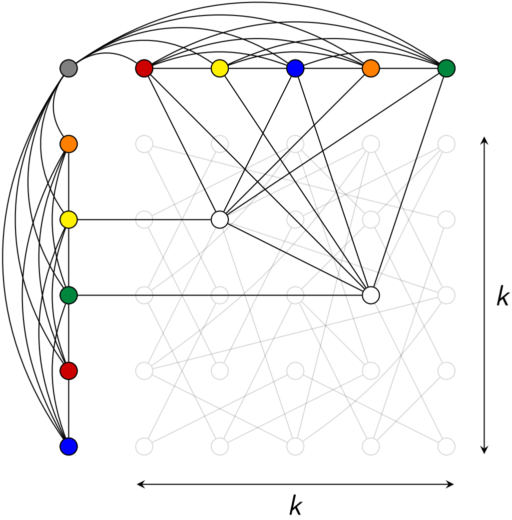
The obtained graph has chromatic number $k+1$ iff the PERMUTATION $k \times k$ CLIQUE instance is positive.

Chromatic Number parameterized by treewidth



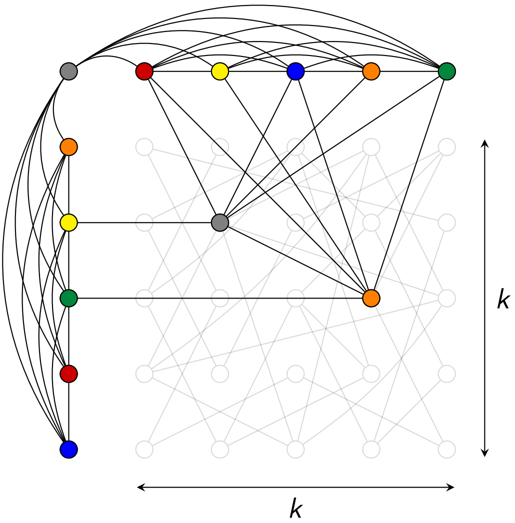
The obtained graph has chromatic number $k+1$ iff the Permutation $k \times k$ Clique instance is positive.

CHROMATIC NUMBER parameterized by treewidth



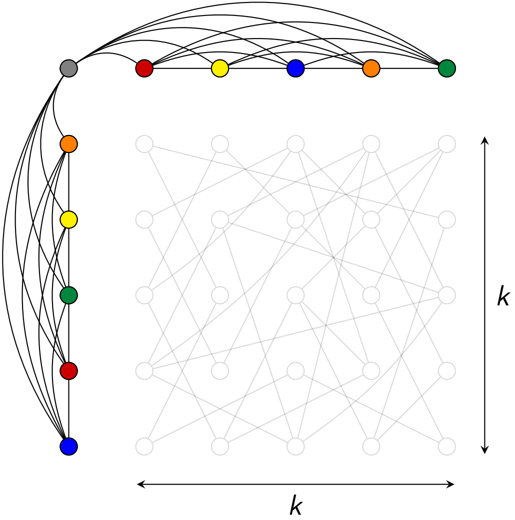
The obtained graph has chromatic number $k+1$ iff the PERMUTATION $k \times k$ CLIQUE instance is positive.

Chromatic Number parameterized by treewidth

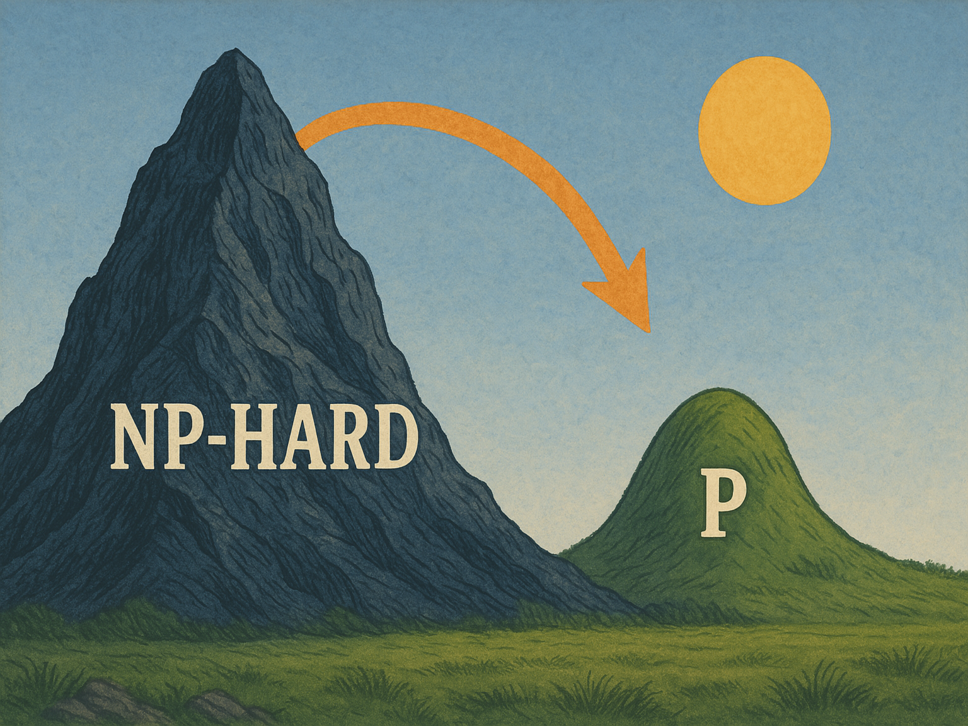


The obtained graph has chromatic number $k + 1$ iff the Permutation $k \times k$ Clique instance is positive.

CHROMATIC NUMBER parameterized by treewidth



ETH $\Rightarrow$ CHROMATIC NUMBER cannot be solved in $2^{o(w \log w)} n^{O(1)}$ time on n-vertex graphs of treewidth at most w.


NP-HARD
P

The three main hypotheses in P

Strong Exponential Time Hypothesis (SETH): $\forall \varepsilon > 0$, $\exists k$ s.t. k-SAT is not in TIME($2^{(1-\varepsilon)n}$).

3-SUM Hypothesis: $\forall \varepsilon > 0$, finding x, y, z s.t. $x + y + z = 0$ in a list of n integers of $[-n^4, n^4]$ is not in TIME($n^{2-\varepsilon}$).

All-Pairs Shortest-Path (APSP) Hypothesis: $\forall \varepsilon > 0$, $\exists c$, APSP with edge weights in $[-n^c, n^c]$ is not in TIME($n^{3-\varepsilon}$).

All these assumptions fail for quantum computers.

The three main hypotheses in P

SETH: SAT is not solvable in 1.99^n.

- k-SAT is solvable in $2^{(1-\Theta(\frac{1}{k}))n}$
- SAT is solvable $2^{(1-\Theta(\frac{1}{\log m/n}))n}$

3-SUM Hypothesis: $3\text{-}\mathrm{SUM}$ is not solvable in $n^{1.99}$

- solvable in $n^2 \frac{(\log \log n)^{O(1)}}{\log^2 n}$ even with real inputs
- linear decision tree with depth $O(n\ log^2 n)$

APSP Hypothesis: APSP is not solvable in $n^{2.99}$

- solvable in cubic time by Floyd-Warshall algorithm
- improved to $n^3/2^{O(\sqrt{\log n})}$

SETH

Introduced in 1999, together with ETH, by Impagliazzo and Paturi

$$\text{SETH} \Rightarrow \text{ETH} \Rightarrow \text{P} \neq \text{NP}$$

- ETH and SETH are then mainly used for NP-hard problems

SETH

Introduced in 1999, together with ETH, by Impagliazzo and Paturi

$$\text{SETH} \Rightarrow \text{ETH} \Rightarrow \text{P} \neq \text{NP}$$

- ETH and SETH are then mainly used for NP-hard problems
- In 2005, SETH is used for the first time for a problem in P

ORTHOGONAL VECTORS,

SETH

Introduced in 1999, together with ETH, by Impagliazzo and Paturi

$$\text{SETH} \Rightarrow \text{ETH} \Rightarrow \text{P} \neq \text{NP}$$

- ETH and SETH are then mainly used for NP-hard problems
- In 2005, SETH is used for the first time for a problem in P
- 2014-, dozens of papers show SETH-hardness of problems in P

Orthogonal Vectors, Diameter, Fréchet Distance, Edit Distance, Longest Common Subsequence, Furthest Pair, dynamic problems, problems from Machine Learning, Model Checking, Language Theory etc.

Reducing from SAT to a problem in P!?

Why no truly subquadratic algorithm for Π was found?

problem SAT
instance φ
n variables

$\xrightarrow[\text{time} \leqslant 1.99^n]{\text{reduction}}$

problem Π
instance I
size $2^{n/2}$

Reducing from SAT to a problem in P!?

Why no truly subquadratic algorithm for Π was found?

$$\begin{array}{c} \text{problem } \mathrm{SAT} \\ \text{instance } \varphi \\ n \text{ variables} \end{array} \xrightarrow[\text{time} \leqslant 1.99^n]{\text{reduction}} \begin{array}{c} \text{problem } \Pi \\ \text{instance } I \\ \text{size } 2^{n/2} \end{array}$$

$1.99^n + (2^{n/2})^{2-\varepsilon} = O(2^{(1-\varepsilon')n})$ for $\varepsilon' := \max(1 - \log 1.99, \frac{\varepsilon}{2}) > 0$

Orthogonal Vectors

Given a set S of N vectors in $\{0,1\}^d$, $\exists u, v \in S$ s.t. $u \cdot v = 0$?

Trivial algorithms in $O(N^2 d)$ and in $O(2^d N)$

Orthogonal Vectors

Given a set S of N vectors in $\{0,1\}^d$, $\exists u, v \in S$ s.t. $u \cdot v = 0$?

Trivial algorithms in $O(N^2 d)$ and in $O(2^d N)$

Theorem (Williams '05)

SETH $\Rightarrow$ Orthogonal Vectors *cannot be solved in* $2^{o(d)} N^{2-\varepsilon}$.

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1										
A_2										
A_3										
A_4										
B_1										
B_2										
B_3										
B_4										

We build $2 \cdot 2^{n/2}$ vectors of $\{0,1\}^{m+2}$

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}$, $x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	0								
A_2										
A_3										
A_4										
B_1										
B_2										
B_3										
B_4										

A_1 assigns red variables

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	0	1							
A_2										
A_3										
A_4										
B_1										
B_2										
B_3										
B_4										

A_1 does *not* satisfy C_1

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	0	1	0						
A_2										
A_3										
A_4										
B_1										
B_2										
B_3										
B_4										

A_1 satisfies C_2

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	0	1	0	0	1	0	0	1	0
A_2										
A_3										
A_4										
B_1										
B_2										
B_3										
B_4										

first vector $(1, 0, 1, 0, 0, 1, 1, 0, 1, 0)$

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	0	1	0	0	1	0	0	1	0
A_2	1	0	0	0	0	1	1	1	0	1
A_3	1	0	0	1	0	1	0	0	1	1
A_4	1	0	0	0	1	1	0	1	1	1
B_1	0	1	1	1	0	0	1	1	1	0
B_2	0	1	0	1	0	1	0	1	0	0
B_3	0	1	1	1	1	1	0	0	0	1
B_4	0	1	0	1	0	0	1	0	0	1

SAT $\longrightarrow$ Orthogonal Vectors

Partition the variables: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- A_i of the red variables and
- B_j of the blue variables

s.t. all the m clauses are satisfied by A_i or by B_j

	R	B	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	1	**0**	1	**0**	**0**	1	**0**	**0**	1	**0**
A_2	1	0	0	0	0	1	1	1	0	1
A_3	1	0	0	1	0	1	0	0	1	1
A_4	1	0	0	0	1	1	0	1	1	1
B_1	0	1	1	1	0	0	1	1	1	0
B_2	0	1	0	1	0	1	0	1	0	0
B_3	0	1	1	1	1	1	0	0	0	1
B_4	**0**	1	**0**	1	**0**	**0**	1	**0**	**0**	1

Diameter

$\mathrm{diam}(G) =$ largest distance between a pair of vertices of G

- In weighted graphs, nothing known better than APSP
- In unweighted graphs, solvable in $\widetilde{O}(n^{\omega})$
- In unweighted sparse graphs, solvable in $O(n^2)$
- In sparse graphs, $\frac{3}{2}$-approximable in $\widetilde{O}(n^{1.5})$

Diameter

diam(G) = largest distance between a pair of vertices of G

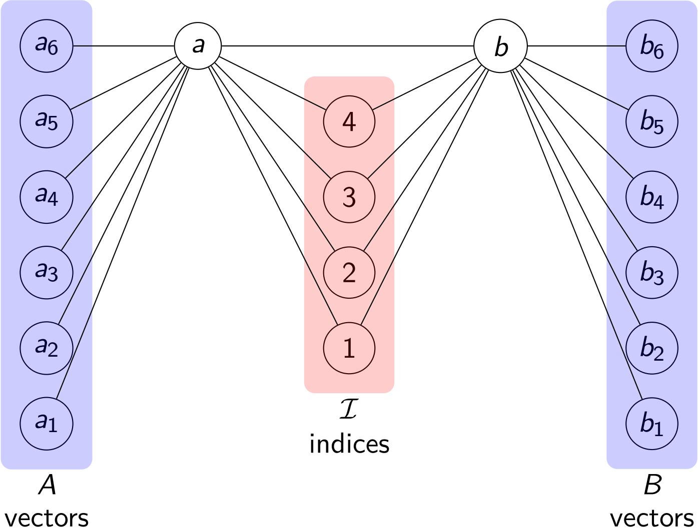
- In weighted graphs, nothing known better than APSP
- In unweighted graphs, solvable in $\widetilde{O}(n^\omega)$
- **In unweighted sparse graphs, solvable in $O(n^2)$**
- **In sparse graphs, $\frac{3}{2}$-approximable in $\widetilde{O}(n^{1.5})$**

Theorem (Roditty–V. Williams '13)

SETH $\Rightarrow \forall \varepsilon > 0$, *$(\frac{3}{2} - \varepsilon)$-approximating sparse, unweighted n-vertex* Diameter *requires $n^{2-o(1)}$.*

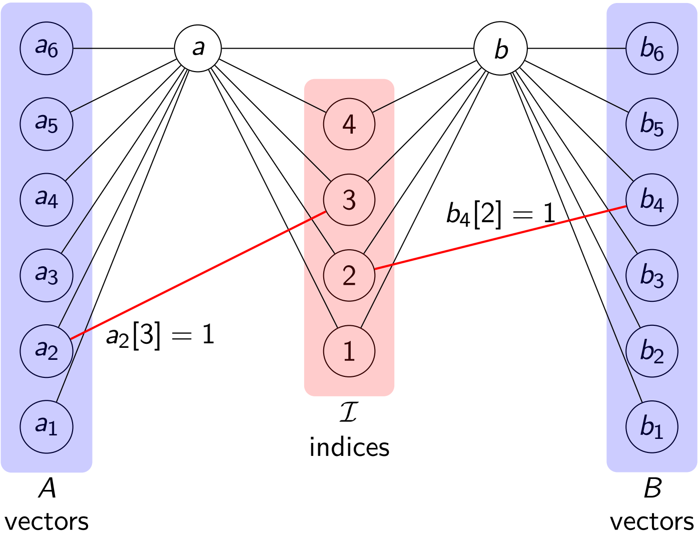
N-vector Orthogonal Vectors $\longrightarrow$ $O(N)$-vertex Diameter

Orthogonal Vectors ⟶ Diameter



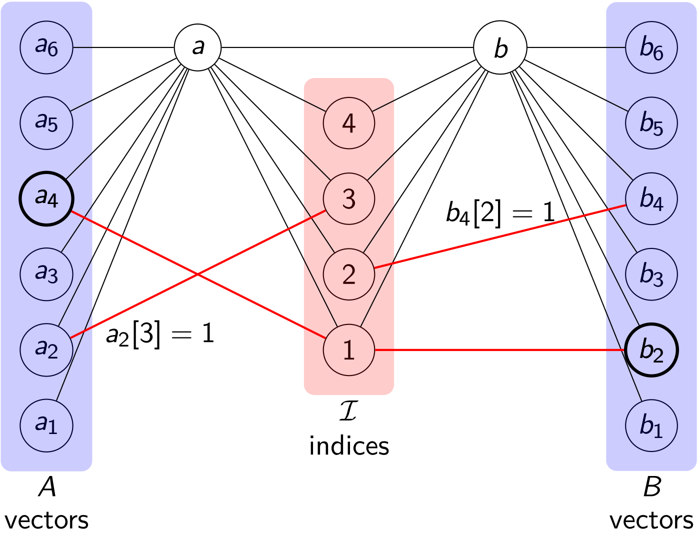
So far, all the pairs but those of $A \times B$ are at distance $\leqslant 2$

Orthogonal Vectors ⟶ Diameter



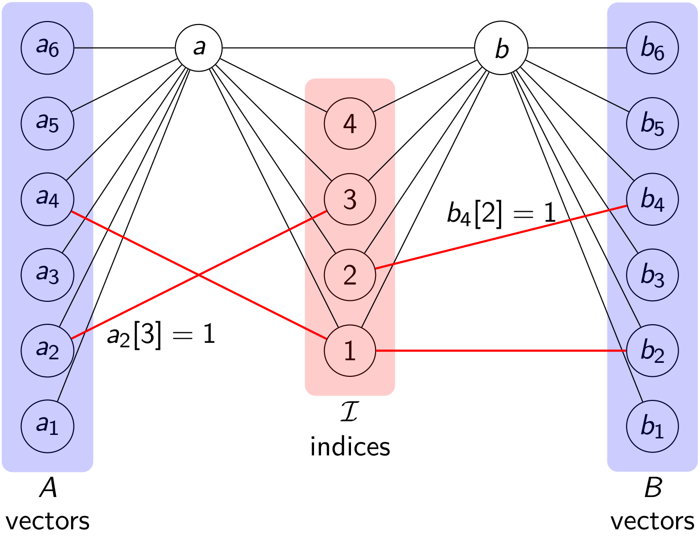
we put an edge between vector v and index i iff $v[i] = 1$

Orthogonal Vectors $\longrightarrow$ Diameter



A pair a_4, b_2 is at distance 2 $\Leftrightarrow a_4 \cdot b_2 \neq 0$

Orthogonal Vectors $\longrightarrow$ Diameter



$\mathrm{diam}(G) = 3 \Leftrightarrow \exists a_i, b_j$ at distance 3 $\Leftrightarrow$ orthogonal pair

3-SUM Hardness

Introduced in 1995 by Gajentaan and Overmars to explain why **some geometric problems require quadratic time**

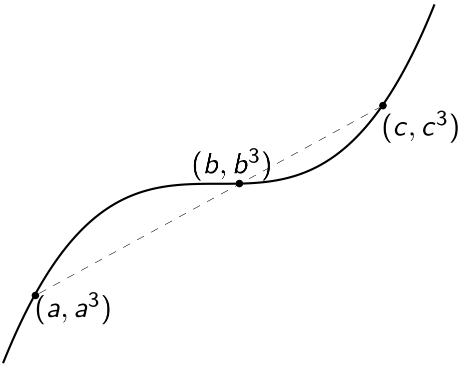
- Given a point set, are there three aligned points?
- computing the area of a union of triangles
- Is there a hole in a union of triangles?
- Is a rectangle covered by a set of infinite strips?
- Is there a line separating (parallel) segments?
- motion planning problems
- visibility problems

3-SUM $\longrightarrow$ 3 Collinear Points

Each integer x is mapped to the point (x, x^3)

3-SUM $\longrightarrow$ 3 Collinear Points

Each integer x is mapped to the point (x, x^3)



If a, b, c are pairwise distinct

$a + b + c = 0 \Leftrightarrow (a, a^3), (b, b^3), (c, c^3)$ are aligned

APSP Hardness

Introduced by Williams and Vassilevska Williams in 2010

APSP is in TIME($n^{3-\varepsilon}$) iff so is one of:

- finding a triangle with negative weight
- finding the diameter or radius of a weighted graph
- Does a given matrix represent a metric?
- finding a shortest cycle in a graph with non-negative weights
- $(min, +)$ matrix multiplication
- computing the Wiener index of a weighted graph
- betweenness centrality of a vertex in a weighted graph

The hypothesis of weighted problems
Unweighted APSP can be solved in time $\widetilde{O}(n^{\omega})$

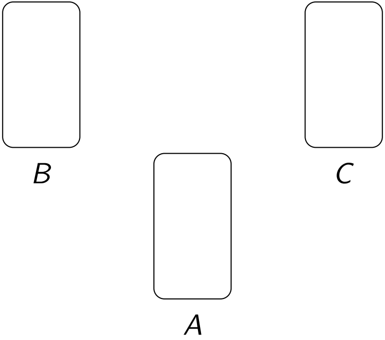
$\mathrm{APNT} \longrightarrow$ Negative Triangle: a Turing reduction

All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?

One can show that APSP and APNT are equivalent

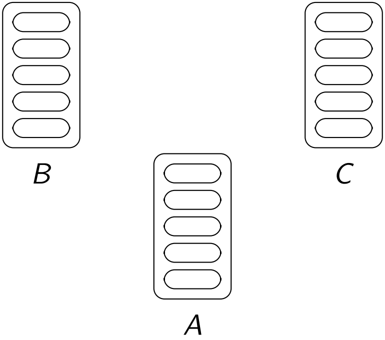
APNT $\longrightarrow$ NEGATIVE TRIANGLE: a Turing reduction

All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



APNT $\longrightarrow$ Negative Triangle: a Turing reduction

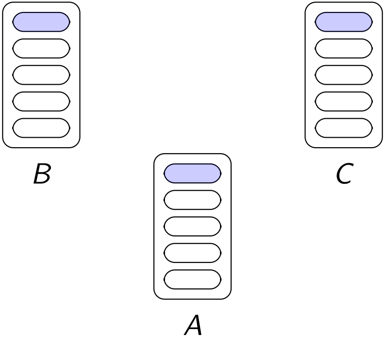
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Partition A, B, C (size n) into $t = n^{2/3}$ groups of size $n/t = n^{1/3}$

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

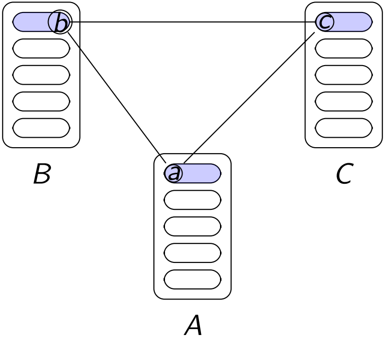
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



For each triple of classes, call Negative Triangle

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

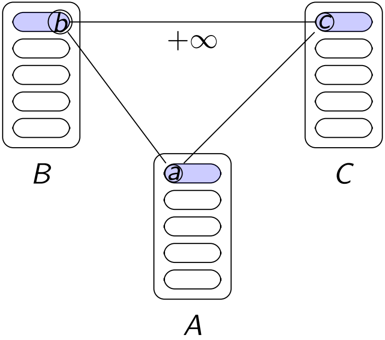
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Write down that the pair bc is satisfied by a and remove bc

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

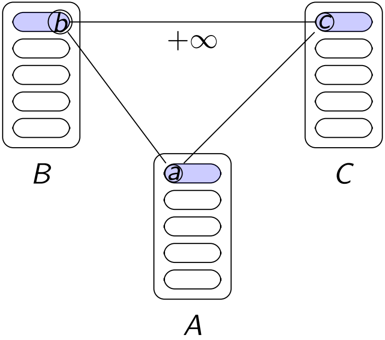
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Write down that the pair bc is satisfied by a and remove bc

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

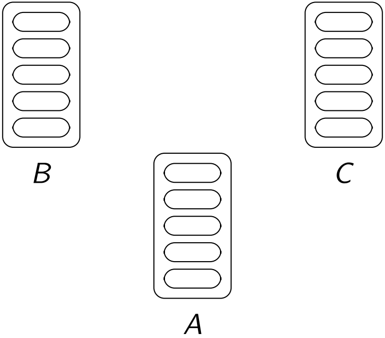
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Continue with the same triple of classes while possible

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

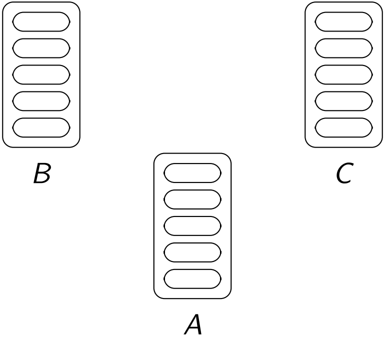
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Report if all the pairs of $B \times C$ were satisfied

APNT $\longrightarrow$ NEGATIVE TRIANGLE: a Turing reduction

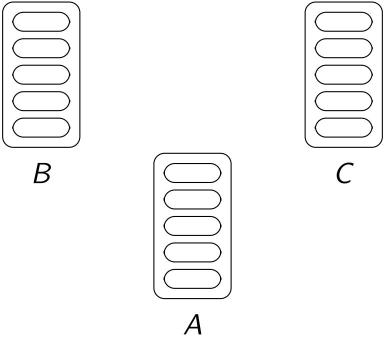
All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there,
$\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Number of calls to NEGATIVE TRIANGLE: $\leqslant n^2 + t^3 = O(n^2)$

APNT $\longrightarrow$ Negative Triangle: a Turing reduction

All-Pairs Negative Triangle (APNT):
Given a tripartite complete graph on (A, B, C), is there, $\forall b \in B, c \in C$, a vertex $a \in A$ such that abc is a negative triangle?



Subinstance size: $3n/t = O(n^{\frac{1}{3}})$, thus $O(n^2 \cdot (n^{\frac{1}{3}})^{3-\varepsilon}) = O(n^{3-\frac{\varepsilon}{3}})$

Things we did not talk about

- SETH lower bounds for parameterized problems
- fine-grained approximability, Gap-ETH (and its bypass)
- NSETH and (weak) evidence against reducing SETH, 3-SUM-H, APSP-H
- no strongly subquadratic algorithm under APSP-H
- Circuit variants of ETH and SETH

Things we did not talk about

- SETH lower bounds for parameterized problems
- fine-grained approximability, Gap-ETH (and its bypass)
- NSETH and (weak) evidence against reducing SETH, 3-SUM-H, APSP-H
- no strongly subquadratic algorithm under APSP-H
- Circuit variants of ETH and SETH

Thank you for your attention!