Et si SAT était vraiment difficile? Quelques conséquences des hypothèses ETH et SETH

Bruno Escoffier, LIP6, Sorbonne Université

ALEA DAYS 2023

March 2023

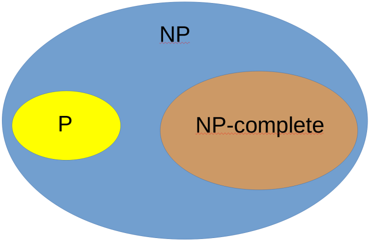
But interesting (at least to me) and

somehow fundamental questions

Classical dichotomy

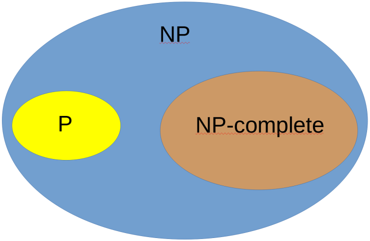
Classical dichotomy

"There are two kinds of people in this world, my friend. Those who have guns and those who do not."



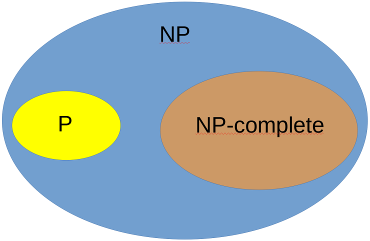
"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not

"There are two kinds of people in this world, my friend. Those who have guns and those who do not."



"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not ... if $P \neq NP$."

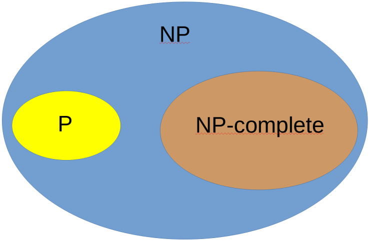
Classical dichotomy



“There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not . . . if $P \neq NP$.”

Well this is nice . . . but not very precise!

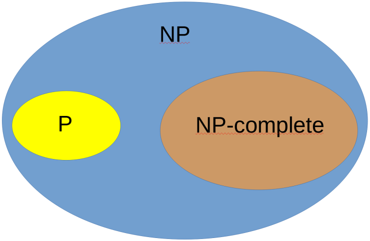
Classical dichotomy



"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not . . . if $P \neq NP$."

Well this is nice . . . but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!



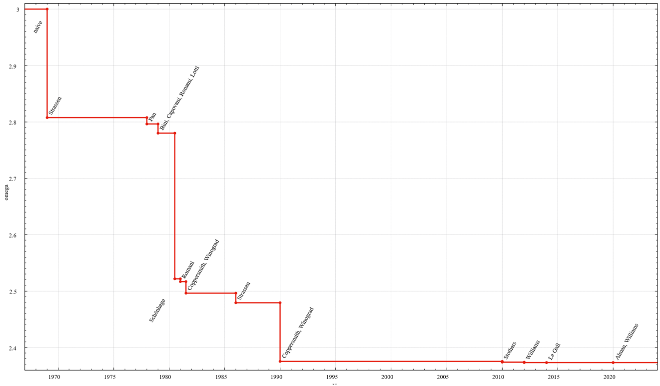
"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not …if $P \neq NP$."

Well this is nice …but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!
  $\rightarrow$ Algorithm design: try to reduce the complexity.

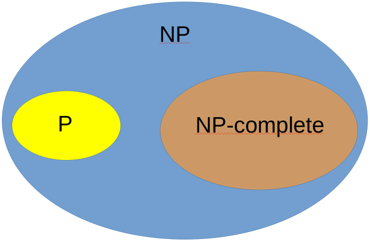
Classical dichotomy

Matrix multiplication: from $O(n^3)$ to $O(n^{2.3728596})$



(source wikipedia)

Classical dichotomy

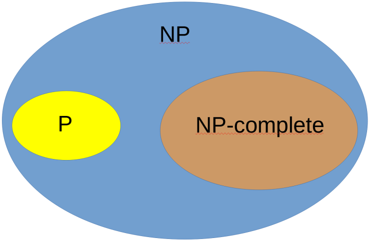


"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not ... if $P \neq NP$."

Well this is nice ... but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!
  $\rightarrow$ Algorithm design: try to reduce the complexity.

Classical dichotomy

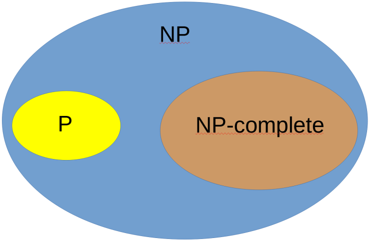


"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not ... if $P \neq NP$."

Well this is nice ... but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!
  $\rightarrow$ Algorithm design: try to reduce the complexity.
  But what about lower bounds??

Classical dichotomy

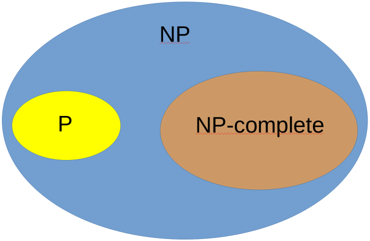


"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not ... if $P \neq NP$."

Well this is nice ... but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!
  $\rightarrow$ Algorithm design: try to reduce the complexity.
  But what about lower bounds??
- NP-complete problems: typically solvable in $O(c^n)$.
  $\rightarrow$ Can't we do better?

Classical dichotomy



"There are two kinds of problems in this world, my friend. Those who have polytime algorithms and those who do not …if $P \neq NP$."

Well this is nice …but not very precise!

- An $O(n^5)$ algorithm is *not* the same as a linear one!
  $\rightarrow$ Algorithm design: try to reduce the complexity.
  But what about lower bounds??
- NP-complete problems: typically solvable in $O(c^n)$.
  $\rightarrow$ Can't we do better?

Find lower bounds, using a stronger hypothesis …
…on Sat! ($P \neq NP \Leftrightarrow \text{Sat} \notin P$)

Sat, k-Sat, ETH and SETH

Sat

- t boolean variables $(x_1, \dots, x_t)$
- m clauses $C_1, \dots, C_m$ $(C_1 = (x_2 \vee \overline{x_4} \vee x_5), \dots)$
- Is there a truth value which satisfies all clauses?

k-Sat: every clause has (exactly/at most) k literals.

Sat, k-Sat, ETH and SETH

Sat

- t boolean variables $(x_1, \ldots, x_t)$
- m clauses $C_1, \ldots, C_m$ $(C_1 = (x_2 \vee \overline{x_4} \vee x_5), \ldots)$
- Is there a truth value which satisfies all clauses?

k-Sat: every clause has (exactly/at most) k literals.

What can we say about Sat/k-Sat?

$\rightarrow$ Solvable in $2^t poly(t, m) = O^*(2^t)$.
$\rightarrow$ Can we do better?

Sat, k-Sat, ETH and SETH

Sat

- t boolean variables $(x_1, \ldots, x_t)$
- m clauses $C_1, \ldots, C_m$ $(C_1 = (x_2 \vee \overline{x_4} \vee x_5), \ldots)$
- Is there a truth value which satisfies all clauses?

k-Sat: every clause has (exactly/at most) k literals.

What can we say about Sat/k-Sat?

$\rightarrow$ Solvable in $2^t poly(t, m) = O^*(2^t)$.
$\rightarrow$ Can we do better? yes and no ...

3-Sat

$$C = (x_1 \vee x_2 \vee \overline{x_3}) \quad \rightarrow \quad \text{only 7 possibilities}$$

$$\text{(all but } x_1 = x_2 = F,\ x_3 = T)$$

$$\rightarrow \quad \text{test all of them}$$

$$\rightarrow \quad T(t) = 7T(t-3)$$

Sat, k-Sat, ETH and SETH

3-Sat

$$
\begin{array}{rcl}
C = (x_1 \vee x_2 \vee \overline{x_3}) & \rightarrow & \text{only 7 possibilities} \\
 & & (\text{all but } x_1 = x_2 = F,\ x_3 = T) \\
 & \rightarrow & \text{test all of them} \\
 & \rightarrow & T(t) = 7T(t-3) \\
 & & (\text{instead of } T(t) = 8T(t-3) \\
 & & \text{for exhaustive search})
\end{array}
$$

3-Sat solvable in $O^*(c_3^t)$, with $c_3 = 7^{1/3} = 1.9... < 2$.

k-Sat

$$
\begin{array}{lll}
C \text{ of size } k & \rightarrow & \text{only } 2^k - 1 \text{ possibilities} \\
 & \rightarrow & \text{test all of them} \\
 & \rightarrow & T(t) = (2^k - 1)T(t-k) \\
 & & \text{(instead of } T(t) = 2^k T(t-k) \\
 & & \text{for exhaustive search)}
\end{array}
$$

k-Sat solvable in $O^*(c_k^t)$, with $c_k = (2^k - 1)^{1/k} < 2$.

Sat, k-Sat, ETH and SETH

Sat

- t boolean variables $(x_1, \ldots, x_t)$
- m clauses $C_1, \ldots, C_m$ $(C_1 = (x_2 \vee \overline{x_4} \vee x_5), \ldots)$
- Is there a truth value which satisfies all clauses?

k-Sat: every clause has (exactly/at most) k literals.

What can we say about Sat/k-Sat?

$\rightarrow$ Solvable in $2^t poly(t, m) = O^*(2^t)$.
$\rightarrow$ Can we do better? yes and no ...
"Yes" for-k Sat.
Significantly better? Subexponential (in t) time?
And for Sat?

Sat, k-Sat, ETH and SETH

Subexponential time? Seems very hard to get, even for 3-Sat $\rightarrow$ ETH.

Sat, k-Sat, ETH and SETH

Subexponential time? Seems very hard to get, even for 3-Sat $\rightarrow$ ETH.

Definition

Let $\mu_k = \inf\{c \geq 0 : k\text{-Sat solvable in } O^*(2^{ct})\}$.

$\mu_k > 0 \rightarrow$ exponential time is needed.

ETH - Exponential Time Hypothesis (Impagliazzo, Paturi, Ramamohan (1999))

$\mu_3 > 0$.

Sat, k-Sat, ETH and SETH

Subexponential time? Seems very hard to get, even for 3-Sat $\rightarrow$ ETH.

Definition

Let $\mu_k = \inf\{c \geq 0 : k\text{-Sat solvable in } O^*(2^{ct})\}$.

$\mu_k > 0 \rightarrow$ exponential time is needed.

ETH - Exponential Time Hypothesis (Impagliazzo, Paturi, Ramamohan (1999))

$\mu_3 > 0$.

And for Sat? No $O^*(2^{ct})$ algorithm with $c < 1$ is known!

SETH - Strong Exponential Time Hypothesis

$\mu_k \rightarrow_{k\rightarrow\infty} 1$.

1. Introduction, ETH and SETH
2. Lower bounds for NP-hard problems
    1. Subexponential time
    2. Parameterized complexity
3. Lower bounds for polynomial problems
4. Concluding remarks

Lower bounds for hard problems: subexponential time

ETH $\rightarrow$ 3-Sat non solvable in subexponential time (wrt t=# variables).

Question

Can we show exponential lower bounds under ETH?

Independent set

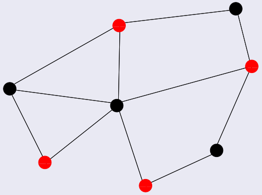


Figure: Indep. set: set of pairwise non adjacent vertices

- Input: (G, k)
  Question: $\alpha(G) \geq k$?

Lower bounds for hard problems: subexponential time

Independent set

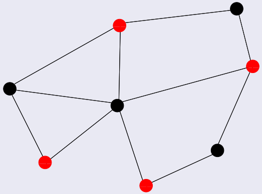
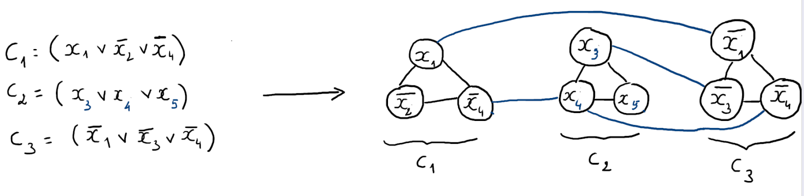


Figure: Indep. set: set of pairwise non adjacent vertices

- Input: (G, k)
  Question: $\alpha(G) \geq k$?
- Solvable in $O^*(2^n)$ ($n = \#$ vertices)
  $\rightarrow$ Not in subexponential time, under ETH?

Reduction 3-Sat ≤ Independent Set

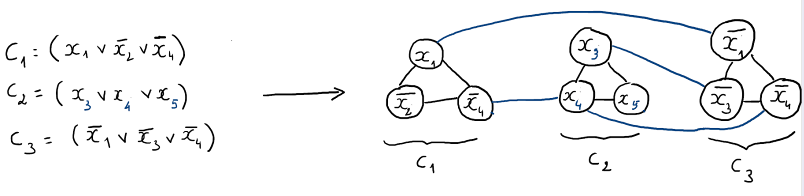


I satisfiable iff $\alpha(G(I)) \geq m$

$\rightarrow$ lower bound $2^{\epsilon n}$ for Independent Set (under ETH)?

Lower bounds for hard problems: subexponential time

Reduction 3-Sat $\leq$ Independent Set



I satisfiable iff $\alpha(G(I)) \geq m$

$\rightarrow$ lower bound $2^{\epsilon n}$ for Independent Set (under ETH)?

No! (well, not yet)

$G(I)$ has $n = 3m$ vertices. $2^{o(n)}$ for IS does *not* give a $2^{o(t)}$ for 3-Sat (contradicting ETH), but a $2^{o(m)}$ (NOT contradicting ETH (yet)).

Lower bounds for hard problems: subexponential time

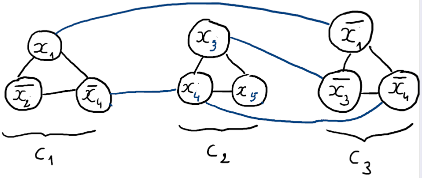
Reduction 3-Sat $\leq$ Independent Set

$C_1 : (x_1 \vee \bar{x}_2 \vee \bar{x}_4)$

$C_2 : (x_3 \vee x_4 \vee x_5)$

$C_3 : (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$

$\longrightarrow$



I satisfiable iff $\alpha(G(I)) \geq m$

$\rightarrow$ lower bound $2^{\epsilon n}$ for Independent Set (under ETH)?

No! (well, not yet)

$G(I)$ has $n = 3m$ vertices. $2^{o(n)}$ for IS does *not* give a $2^{o(t)}$ for 3-Sat (contradicting ETH), but a $2^{o(m)}$ (NOT contradicting ETH (yet)).

We need a reduction where $n = O(t)$ ... or to work with 3-Sat instances with $m = O(t)$ clauses.

Sparsification lemma (Impagliazzo et al. (2001))

Let 3-Sat(B) be the restriction of 3-Sat to instances where $m \leq Bt$.
ETH holds iff $\exists B$ such that "it holds for 3-Sat(B)"

Sparsification lemma (Impagliazzo et al. (2001))

Let 3-Sat(B) be the restriction of 3-Sat to instances where $m \leq Bt$.
ETH holds iff $\exists B$ such that "it holds for 3-Sat(B)"

3-Sat $\rightarrow$ 3-Sat(B) $\rightarrow$ Independent Set

Lower bounds for hard problems: subexponential time

Sparsification lemma (Impagliazzo et al. (2001))

Let 3-Sat(B) be the restriction of 3-Sat to instances where $m \leq Bt$.
ETH holds iff $\exists B$ such that "it holds for 3-Sat(B)"

$$\text{3-Sat} \rightarrow \text{3-Sat}(B) \rightarrow \text{Independent Set}$$

Hardness of Independent Set

Under ETH, there exists $\epsilon > 0$ such that Independent is not solvable in $2^{\epsilon n}$ (with $n = \#$ vertices).

And the same for many other problems (3-colorability, Hamiltonian path,...)

Lower bounds for hard problems: subexponential time

Question

Can we show exponential lower bounds under ETH?

Answer

Yes we can

. . . well, this was expected, but it was not that direct

By the way: shall we buy (S)ETH?

By the way: shall we buy (S)ETH?

Reminder

Under ETH, there exists $\epsilon > 0$ such that Independent is not solvable in $2^{\epsilon n}$ (with $n = \#$ vertices).

Is the reverse true? Also for other problems?

By the way: shall we buy (S)ETH?

Reminder

Under ETH, there exists $\epsilon > 0$ such that Independent is not solvable in $2^{\epsilon n}$ (with $n = \#$ vertices).

Is the reverse true? Also for other problems?
$\rightarrow$ Yes: if ETH fails, then many well known optimization problems would be solvable in subexponential time.

By the way: shall we buy (S)ETH?

Reminder

Under ETH, there exists $\epsilon > 0$ such that Independent is not solvable in $2^{\epsilon n}$ (with $n = \#$ vertices).

Is the reverse true? Also for other problems?
$\rightarrow$ Yes: if ETH fails, then many well known optimization problems would be solvable in subexponential time.

Shall we buy SETH? ($\Rightarrow$ no c^t algo for Sat with $c < 2$)

By the way: shall we buy (S)ETH?

Reminder

Under ETH, there exists $\epsilon > 0$ such that Independent is not solvable in $2^{\epsilon n}$ (with $n = \#$ vertices).

Is the reverse true? Also for other problems?
$\rightarrow$ Yes: if ETH fails, then many well known optimization problems would be solvable in subexponential time.

Shall we buy SETH? ($\Rightarrow$ no c^t algo for Sat with $c < 2$)
Well...

...but at least you should compete for the Godel prize if you disprove it!

1. Introduction, ETH and SETH
2. Lower bounds for NP-hard problems
    1. Subexponential time
    2. Parameterized complexity
3. Lower bounds for polynomial problems
4. Concluding remarks

Independent set

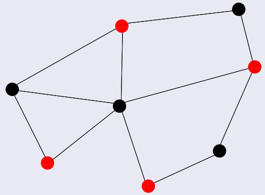


Figure: Indep. set: set of pairwise non adjacent vertices

- Input: (G, k)
  Question: $\alpha(G) \geq k$?

Independent set

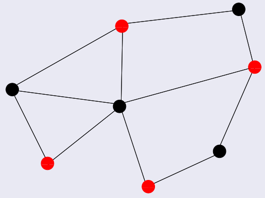


Figure: Indep. set: set of pairwise non adjacent vertices

- Input: (G, k)
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? Get rid of it?

Independent set

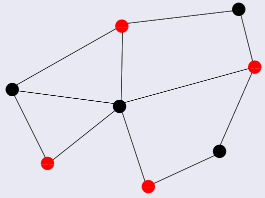


Figure: Indep. set: set of pairwise non adjacent vertices

- Input: (G, k)
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? Get rid of it? Parameterized complexity.

Lower bounds for hard problems: paramaterized complexity

Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? Get rid of it?
  Parameterized complexity.

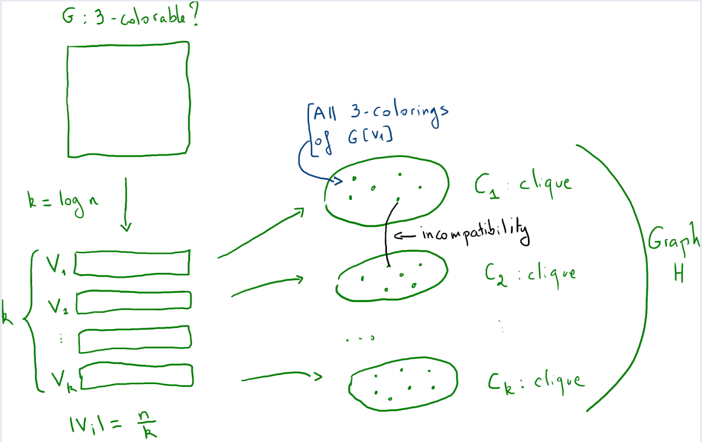
Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).

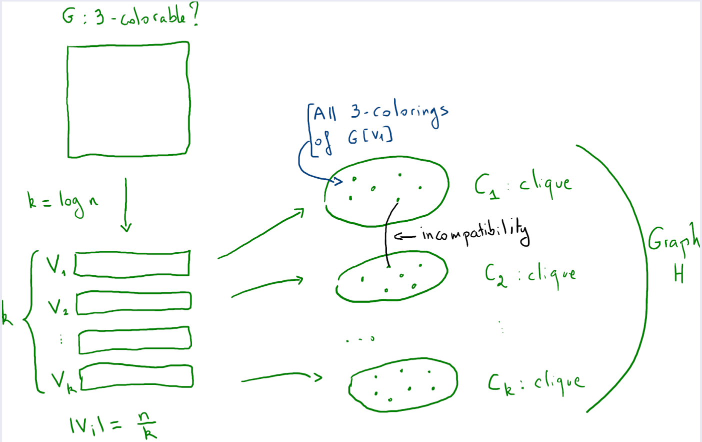
Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).
  In $2^k n^{\sqrt{k}}$? Or at least $f(k)n^{o(k)}$?

If I.S. solvable in $O(n^{o(k)})$ then 3-coloring solvable in $O(2^{o(n)})$.

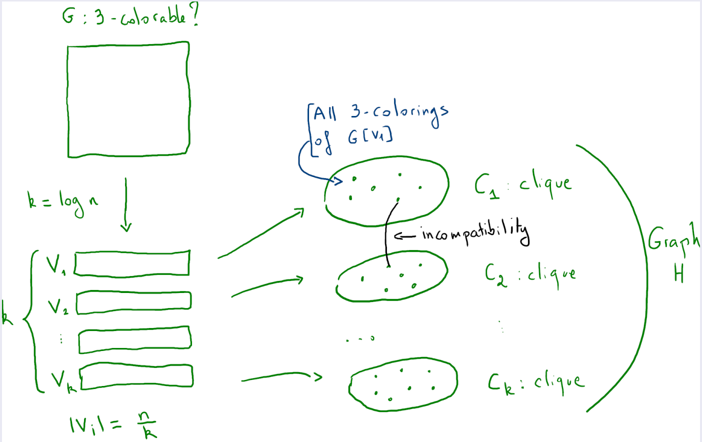


If I.S. solvable in $O(n^{o(k)})$ then 3-coloring solvable in $O(2^{o(n)})$.



G 3-colorable iff $\alpha(H) = k$.

If I.S. solvable in $O(n^{o(k)})$ then 3-coloring solvable in $O(2^{o(n)})$.



G 3-colorable iff $\alpha(H) = k$.

- $|C_i| \leq 3^{n/k} \rightarrow H$ has $N \leq k3^{n/k}$ vertices.
- $\alpha(H) = k$? Time $N^{o(k)} \leq k^{o(k)} 3^{n.o(k)/k} = 2^{o(n)}$ $(k = \log n)$.

Lower bound

Under ETH, I.S. is not solvable in $O(n^{o(k)})$

Lower bound

Under ETH, I.S. is not solvable in $O(n^{o(k)})$ and not in $f(k)n^{o(k)}$, for any function f. (Chen, Chor, Fellows, Huang, Juedes, Kanj, and Xia (2005))

The same occurs for other problems (e.g., dominating set).

Lower bound

Under ETH, I.S. is not solvable in $O(n^{o(k)})$ and not in $f(k)n^{o(k)}$, for any function f. (Chen, Chor, Fellows, Huang, Juedes, Kanj, and Xia (2005))

The same occurs for other problems (e.g., dominating set).
$\rightarrow$ Remark: use of non-polytime reduction.

Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).
  In $2^k n^{\sqrt{k}}$? Or at least $f(k)n^{o(k)}$?

Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).
  In $2^k n^{\sqrt{k}}$? Or at least $f(k)n^{o(k)}$?
  $\rightarrow$ No under ETH

Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).
  In $2^k n^{\sqrt{k}}$? Or at least $f(k)n^{o(k)}$?
  $\rightarrow$ No under ETH
  In time n^{ck} for some $c < 1$?

Independent set

- Input: (G, k)
  Parameter: k
  Question: $\alpha(G) \geq k$?
- Solvable in $O(k^2 n^k)$ ($n = \#$ vertices)
  $\rightarrow$ Can we improve the degree of the polynomial? ~~Get rid of it?~~
  Parameterized complexity.
  $\rightarrow$ Not solvable in $f(k)n^c$ (if FPT$\neq$ W[1]).
  In $2^k n^{\sqrt{k}}$? Or at least $f(k)n^{o(k)}$?
  $\rightarrow$ No under ETH
  In time n^{ck} for some $c < 1$?
  $\rightarrow$ Well, doable for IS ... but not for other problems under SETH (even no $n^{k-\epsilon}$).

We can

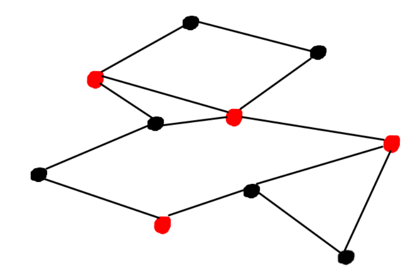
$\rightarrow$ Find lower bounds beyond polytime:

- under ETH (no $2^{o(n)}$, no $n^{o(k)}$, ...),
- under SETH, sharp bounds,

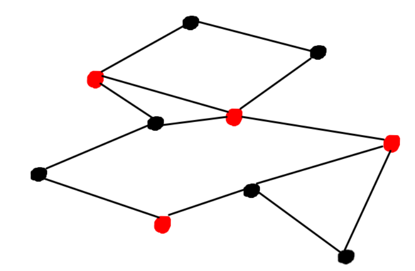
both in classical and parameterized complexity frameworks.

1. Introduction, ETH and SETH
2. Lower bounds for NP-hard problems
   1. Subexponential time
   2. Parameterized complexity
3. Lower bounds for polynomial problems
4. Concluding remarks

Dominating Set

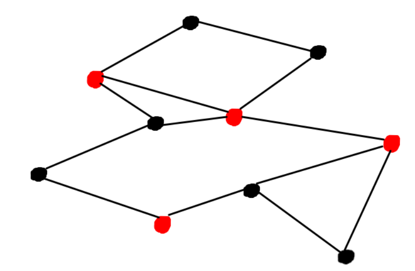


Dominating Set



Dominating Set: S such that every vertex not in S has a neighbor in S.

Dominating Set



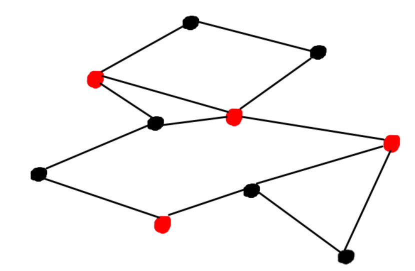
Dominating Set: S such that every vertex not in S has a neighbor in S.

- k-DS: does G has a D.S. of size k?

Enumerating all subsets of size $k \rightarrow n^k$.
Can I avoid this? Can I solve 3-DS in $n^{3-\epsilon}$? k-DS in $n^{k-\epsilon}$ for some/any k?

Lower bounds for polynomial problems

Dominating Set

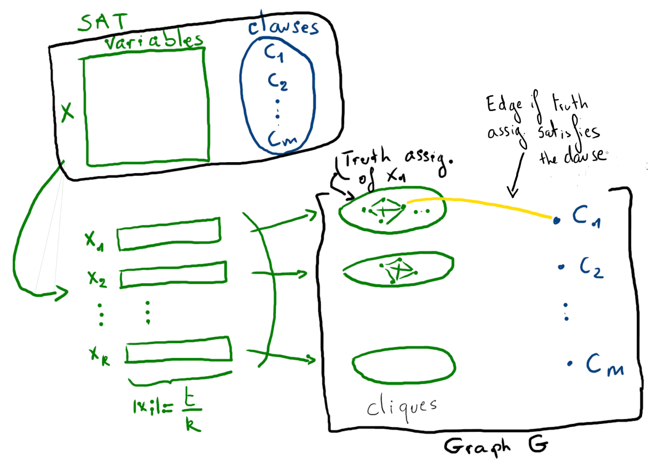


Dominating Set: S such that every vertex not in S has a neighbor in S.

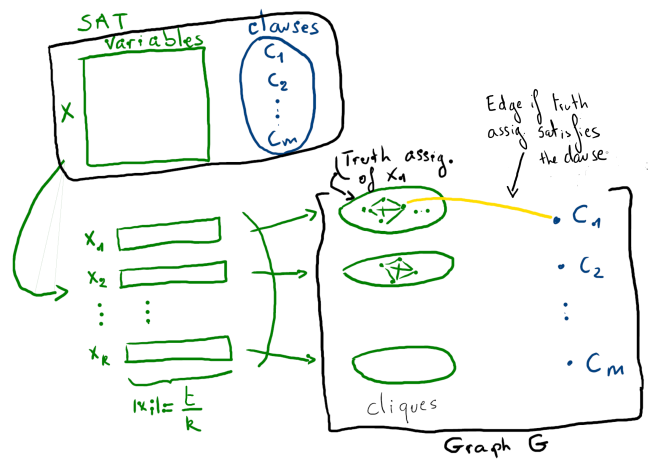
- k-DS: does G has a D.S. of size k?

Enumerating all subsets of size $k \to n^k$.
Can I avoid this? Can I solve 3-DS in $n^{3-\epsilon}$? k-DS in $n^{k-\epsilon}$ for some/any k?
No, under SETH! No $n^{3-\epsilon}$ for 3-DS; $\forall k$, no $n^{k-\epsilon}$ for k-DS!!

$\forall k \geq 3, \epsilon > 0$: if k-DS is solvable in $O(n^{k-\epsilon})$ then SETH is false.

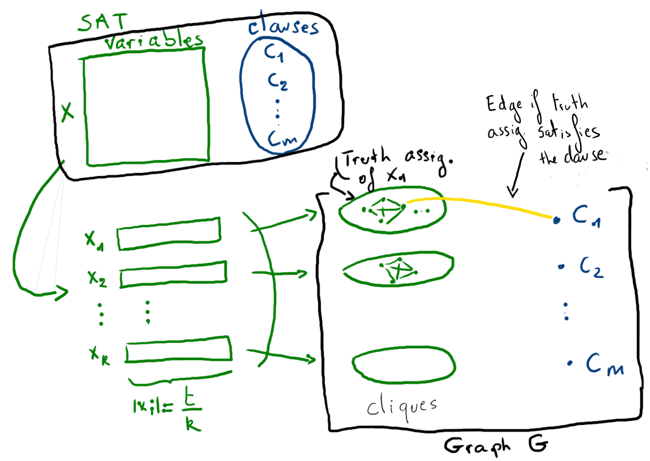


$\forall k \geq 3, \epsilon > 0$: if k-DS is solvable in $O(n^{k-\epsilon})$ then SETH is false.



G has a D.S. of size k iff the formula is satisfiable.

$\forall k \geq 3, \epsilon > 0$: if k-DS is solvable in $O(n^{k-\epsilon})$ then SETH is false.



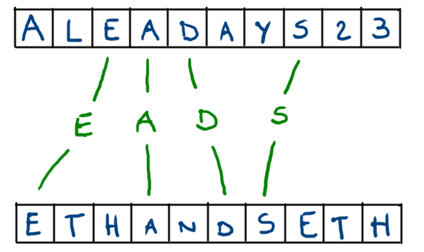
G has a D.S. of size k iff the formula is satisfiable.

G has $n \leq k2^{t/k} + m$ vertices $\rightarrow$ G has a D.S. of size k?

Time $n^{k-\epsilon} \leq 2^{t(1-\epsilon/k)} poly(m,t)$ $\rightarrow$ SETH is false.

Lower bound for DS, also for other classical problems.

LCS (longest common subsequence)

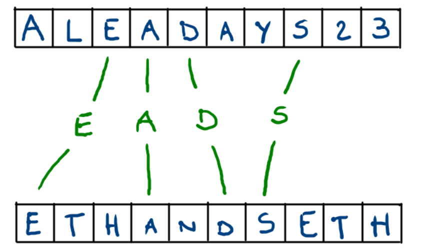


Solvable in $O(n^2)$ ($n = |U| = |W|$) using DP

Lower bounds for polynomial problems

Lower bound for DS, also for other classical problems.

LCS (longest common subsequence)



Solvable in $O(n^2)$ ($n = |U| = |W|$) using DP

Theorem ((Abboud et al. 2015))

Under SETH, $\forall \epsilon > 0$, LCS is not solvable in $O(n^{2-\epsilon})$.

We can

$\rightarrow$ Find lower bounds for polytime problems, under SETH: *fine-grained complexity*.

1. Introduction, ETH and SETH
2. Lower bounds for NP-hard problems
   1. Subexponential time
   2. Parameterized complexity
3. Lower bounds for polynomial problems
4. Concluding remarks

Concluding remarks

Other topics:

- Lower bounds for other problems
- Lower bounds for approximation algorithms
- Randomized ETH
- ...

Lower bounds for other problems?

Back to independent set

Theorem

A graph has either an independent set of size $\lfloor \log_2(n)/2 \rfloor$, or a clique of size $\lfloor \log_2(n)/2 \rfloor$.

Lower bounds for other problems?

Back to independent set

Theorem

A graph has either an independent set of size $\lfloor \log_2(n)/2 \rfloor$, or a clique of size $\lfloor \log_2(n)/2 \rfloor$.

but can we determine which case(s) occur(s)?

Lower bounds for other problems?

Back to independent set

Theorem

A graph has either an independent set of size $\lfloor \log_2(n)/2 \rfloor$*, or a clique of size* $\lfloor \log_2(n)/2 \rfloor$*.*

but can we determine which case(s) occur(s)?

- A graph G
- Does $\alpha(G) \geq \log(n)$?

Back to independent set

Theorem

A graph has either an independent set of size $\lfloor \log_2(n)/2 \rfloor$*, or a clique of size* $\lfloor \log_2(n)/2 \rfloor$*.*

but can we determine which case(s) occur(s)?

- A graph G
- Does $\alpha(G) \geq \log(n)$?

$\rightarrow$ solvable in $O(n^{O(\log n)}) = 2^{\text{polylog } n}$
Not NP-complete (unless NP$\subseteq$ QP)... but seems hard to solve in polytime !

Lower bounds for other problems?

Back to independent set

Theorem

A graph has either an independent set of size $\lfloor \log_2(n)/2 \rfloor$, or a clique of size $\lfloor \log_2(n)/2 \rfloor$.

but can we determine which case(s) occur(s)?

- A graph G
- Does $\alpha(G) \geq \log(n)$?

$\rightarrow$ solvable in $O(n^{O(\log n)}) = 2^{\text{polylog } n}$
Not NP-complete (unless NP$\subseteq$ QP)... but seems hard to solve in polytime !

Theorem

It is not in P, and even not solvable in $n^{o(\log n)}$, under ETH!

Lower bounds for other problems?

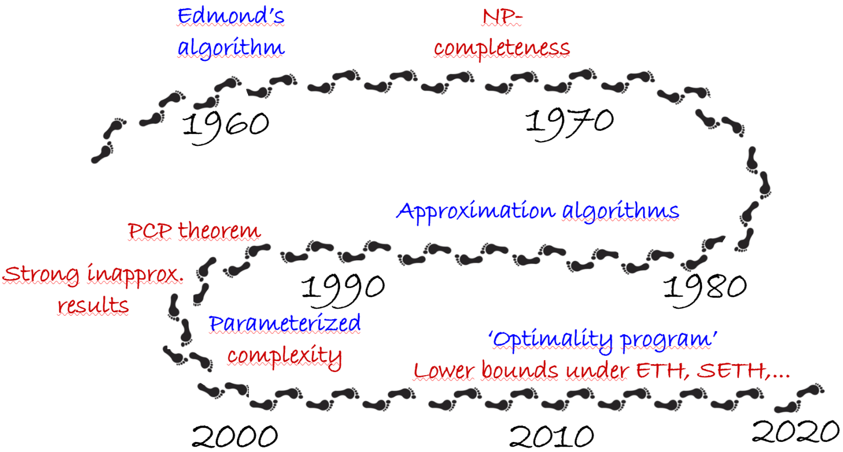
We can

$\rightarrow$ Get hardness results for problems "hard but not NP-complete", under ETH.

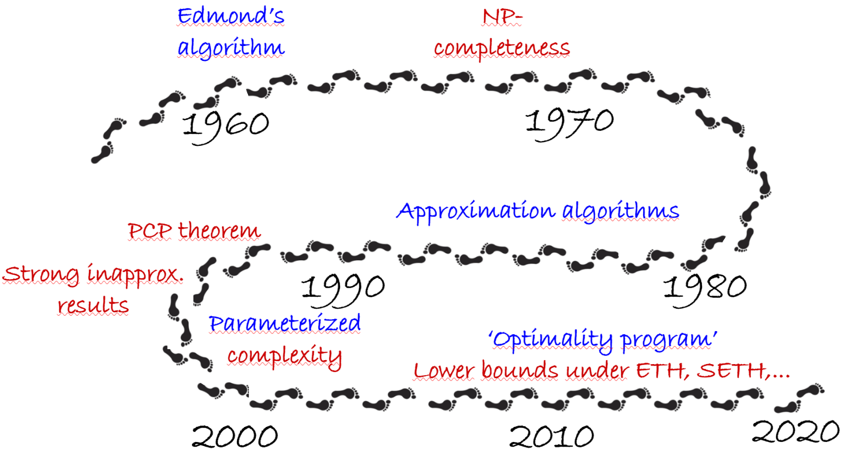
Concluding remarks

- Lower bounds for other problems
- Lower bounds for approximation algorithms
- Randomized ETH
- ...

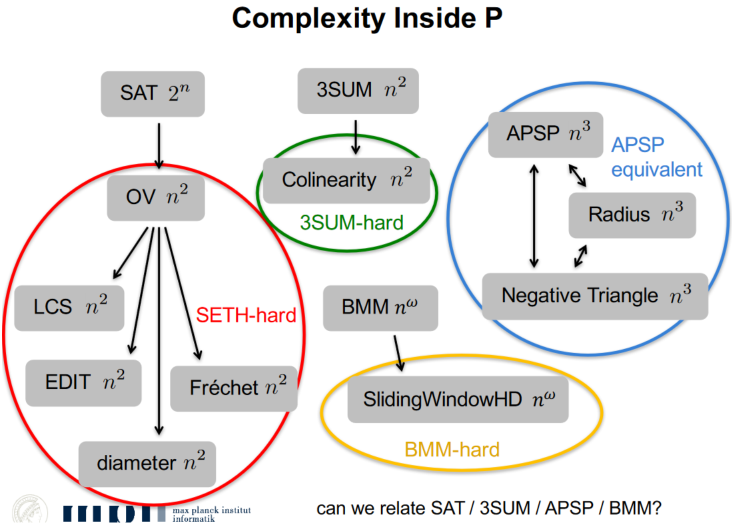
Concluding remarks



Concluding remarks



Merci de votre attention!



From a lecture of Karl Bringmann, `https://www.cs.sbg.ac.at/~forster/courses/polycomp/slides/polycomp11.pdf`.

Some tight results under SETH

Under SETH

- I.S. is not solvable in $(2-\epsilon)^{tw} n^c$ with tw=treewidth (Lokshtanov, Marx, and Saurabh 2010). For D.S.: no $(3-\epsilon)^{tw} n^c$.
- Many tight bounds for other parameters (pathwidth, cliquewidth,...) in parameterized complexity.
- No $(2-\epsilon)^n$ algorithm for hitting set.
- Diameter of a graph, under SETH: no $m^{2-\epsilon}$ (exact) algorithm (Roditty and Williams 2013), no $(2-\epsilon)$-approximation in $m^{1+o(1)}$ (even in sparse graphs), (Li'21, Dalirrooyfard Wei'20)

Subexponential time lower bounds under ETH: There is no $2^{o(\sqrt{n})}$ algorithm for Vertex Cover, 3-Colorability, and Hamiltonian Path for planar graphs.

(In)approximability

In polynomial time:

- $\forall c > 0$: no c-approximation algorithm
- (and even) for all $\epsilon > 0$: no $n^{\epsilon - 1}$-approximation algorithm.

(In)approximability

In polynomial time:

- $\forall c > 0$: no c-approximation algorithm
- (and even) for all $\epsilon > 0$: no $n^{\epsilon-1}$-approximation algorithm.

Under ETH:

- $\forall c > 0$: no $2^{n^{1-\epsilon}}$-time c-approximation algorithm. (Bonnet, Escoffier, Kim, Paschos, 2013)

(In)approximability

In polynomial time:

- $\forall c > 0$: no c-approximation algorithm
- (and even) for all $\epsilon > 0$: no $n^{\epsilon - 1}$-approximation algorithm.

Under ETH:

- $\forall c > 0$: no $2^{n^{1-\epsilon}}$-time c-approximation algorithm. (Bonnet, Escoffier, Kim, Paschos, 2013)
- $\sqrt{n}$-approximation: easy to get in $O^*(2^{\sqrt{n}}) \rightarrow$ subexponential time.

(In)approximability

In polynomial time:

- $\forall c > 0$: no c-approximation algorithm
- (and even) for all $\epsilon > 0$: no $n^{\epsilon-1}$-approximation algorithm.

Under ETH:

- $\forall c > 0$: no $2^{n^{1-\epsilon}}$-time c-approximation algorithm. (Bonnet, Escoffier, Kim, Paschos, 2013)
- $\sqrt{n}$-approximation: easy to get in $O^*(2^{\sqrt{n}}) \rightarrow$ subexponential time. But no better! (The same for other ratios) (Chalermsook, Laekhanukit, Nanongkai, 2013)

Randomized ETH

Definition r-ETH (from Dell et al. 2012)

There is a constant $c > 0$ such that no randomized algorithm can decide 3-Sat in time 2^{ct} with error probability at most $1/3$.

Negative results under r-ETH:

- Computing the permanent of a 0-1 matrix of size $n \times n$ cannot be done in $2^{o(n)}$, and not even in time $2^{o(m)}$ where m is the number of non-zero elements.
- Some (tight) lower bounds for approximation ratios in subexponential time, e.g. in Katsikarelis, Lampis, Paschos 2019.

Also #-ETH: $\exists c$ s.t. counting the number of sat. assignments for 3-SAT cannot be done in 2^{ct}.