Computational complexity: NP completeness

Petr Kurapov

Fall 2024

MIPT

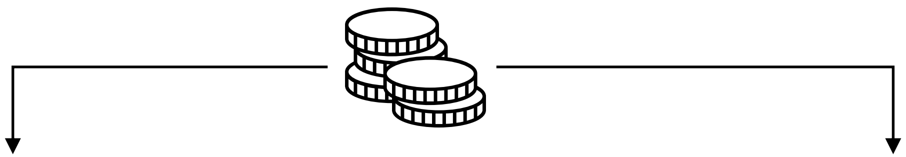
Previous results

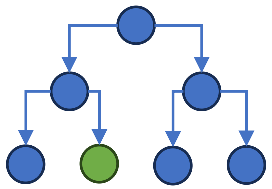
- Math model for computations – Turing machine (TM)
- There's a universal TM that can simulate any other efficiently
- Some functions are not computable by any TM
- Defined class of "easy" problems P (can be solved efficiently)

Complexity class $\boldsymbol{NP}$

Nondeterministic Turing Machine (NDTM) – not physically realizable

$\boldsymbol{NP}$ – those problems that NDTM can solve efficiently (poly)





δ_0

IF			THEN			
Input symbol read	Work/output tape symbol read	Current state	Move input head	New work/output tape symbol	Move work/output tape	New state
A	B	Q	Right	B’	Left	Q’

δ_1

IF			THEN			
Input symbol read	Work/output tape symbol read	Current state	Move input head	New work/output tape symbol	Move work/output tape	New state
A	B	Q	Right	B’	Left	Q’

The sequence of choices can be viewed as a **certificate**

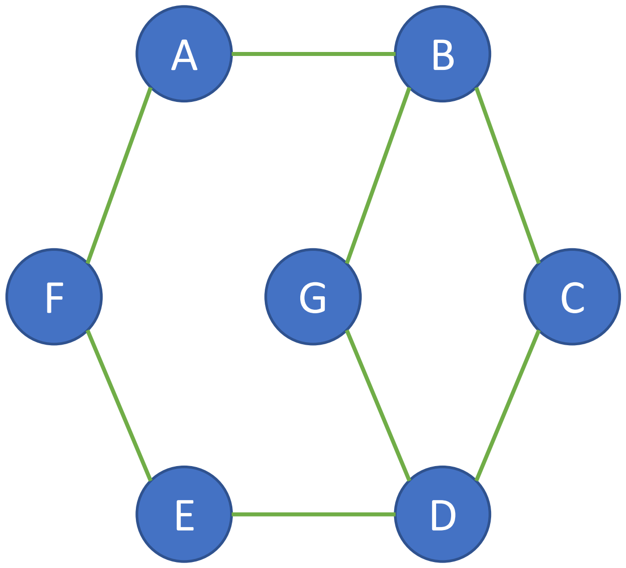
Complexity class $\boldsymbol{NP}$

Efficiently verifiable problems – creative effort is required for the solution, but not for verification

- $L \subseteq \{0,1\}^* \in \boldsymbol{NP}\ if\ \exists p, M_{verifier}\ - poly:\ \forall x$:
- $x \in L \leftrightarrow \exists cert \in \{0,1\}^{p(|x|)}: M(x, cert) = 1$
- P is subset of NP (p can be 0)

Example: Independent Set Problem (ISP)

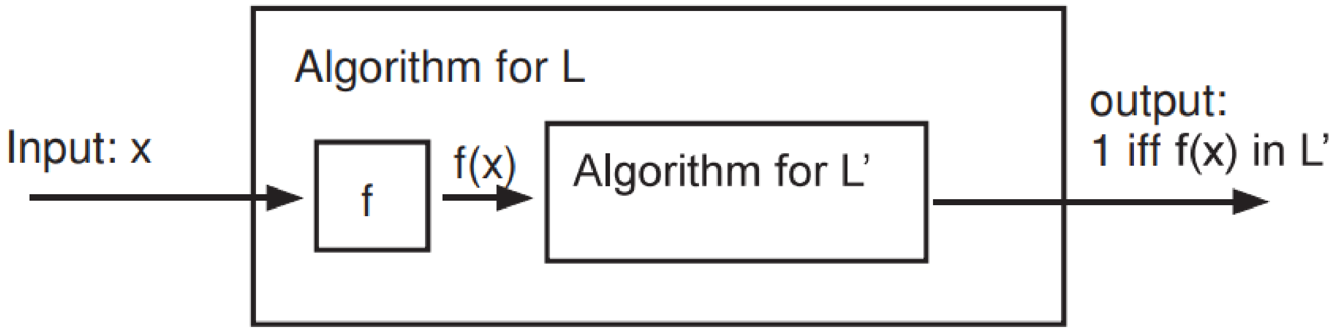
- $ISet = \{(G,k): \exists S \subseteq V_G: |S| \geq k \;\&\; \forall u,v \in S, \overline{uv} \notin E_G\}$
- $\langle G,k \rangle \leftrightarrow \{0,1\}^*$
- $ISP \in NP$



{F,G,C}, {A,E,G,C}

Polynomial reducibility

- A.k.a. many-to-one-reducibility, polynomial-time mapping, polynomial-time Karp reducibility.
- L is reducible to L' ($L \leq_p L'$) if $\exists f(x): \forall x - x \in L \leftrightarrow f(x) \in L'$

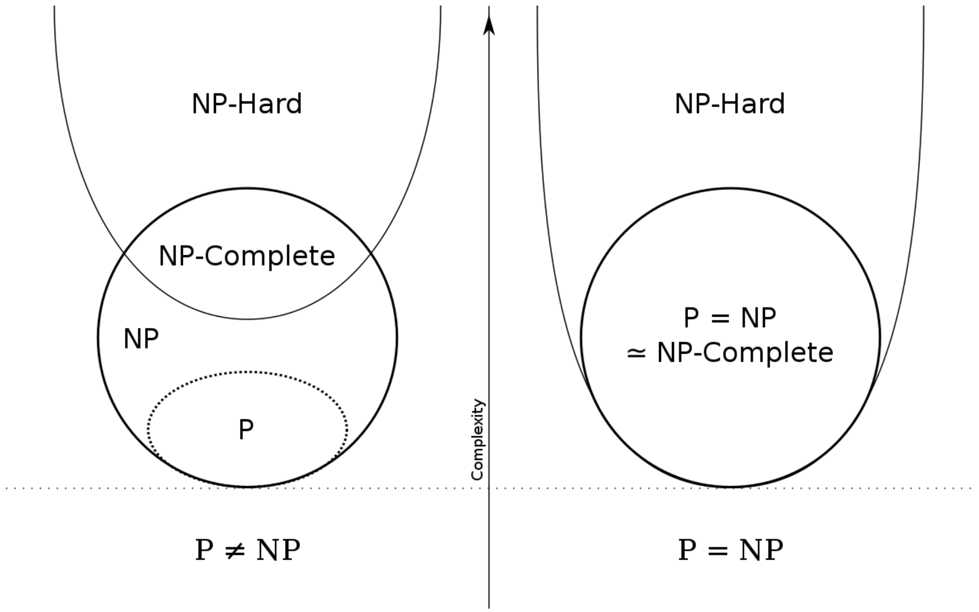


Source: Computational Complexity: A Modern Approach [2]

NP-hard, NP-complete

- L' - NP-hard if $L \leq_p L'$ for $\forall L \in NP$
- L' - NP-complete if it's NP-hard & in NP

- $L \in NP_h \& L \in P \rightarrow P = NP$
- $L \in NP_c \rightarrow L \in P \leftrightarrow P = NP$



Source: wiki at https://en.wikipedia.org/wiki/NP-hardness#:~:text=In%20computational%20complexity%20theory%2C%20NP,is%20the%20subset%20sum%20problem

NP-complete language example*

- $S = \left\{\left\langle a, x, 1^k, 1^t \right\rangle: \; \exists s = \{0,1\}^k \; so \; that \; M_a(x,s) = 1 \; within \; t \; steps\right\}$

Proof:

- $L \in NP \rightarrow p, M: x \in L \; iff \; \exists u \in \{0,1\}^{p(|x|)}, M(x,u) = 1$, runs in q (polynomial) steps (by definition).
- Reduce L to S: map $x \in \{0,1\}^*$ to $\left\langle M_a, x, 1^{p(|x|)}, 1^{q(|x|+p(|x|))} \right\rangle$ - the mapping can be done in polynomial time. The string $\in S$, meaning there's a $M(x,u)$ that yields 1 in $q(|x| + p(|x|))$ steps.

Cook-Levin theorem

- Boolean formula: $z \in \{0,1\}^n, \varphi(z) = f(\bar{u})$
- CNF (Conjunctive Normal Form): $\bigwedge_i (\bigvee_i u_{ij})$
- 3CNF: $(u \vee v \vee w) \wedge (v \vee \bar{w} \vee z) \wedge (\bar{u} \vee v \vee \bar{z})$
- SAT – set of satisfiable CNF
- 3SAT – set of satisfiable 3CNF

- SAT is NP-complete
- 3SAT is NP-complete

S. A. Cook. The complexity of theorem proving procedures. https://www.inf.unibz.it/~calvanese/teaching/11-12-tc/material/cook-1971-NP-completeness-of-SAT.pdf

ISP is NP-complete

- $ISet = \{(G, k): \exists S \subseteq V_G: |S| \geq k \;\&\; \forall u, v \in S, \overline{uv} \notin E_G\}$
- ISet is in NP

ISP is NP-complete

- Transform a 3CNF formula with m clauses to a graph with 7m vertices
- Each vertex represents a variation of a single clause that satisfy it

E.g., $U_2 \cup \overline{U_{17}} \cup U_{26}$

This is a vertex in the graph G
Represents a partial assignment

U_2	0	0	0	1	1	1	1
U_{17}	0	0	1	0	0	1	1
U_{26}	0	1	1	0	1	0	1

Missing (0,1,0) assignment since
it does not satisfy the clause

ISP is NP-complete

E.g., $U_2 \cup \overline{U_{17}} \cup U_{26}$

U_2	0	0	0	1	1	1	1
U_{17}	0	0	1	0	0	1	1
U_{26}	0	1	1	0	1	0	1

These partial assignments are inconsistent!

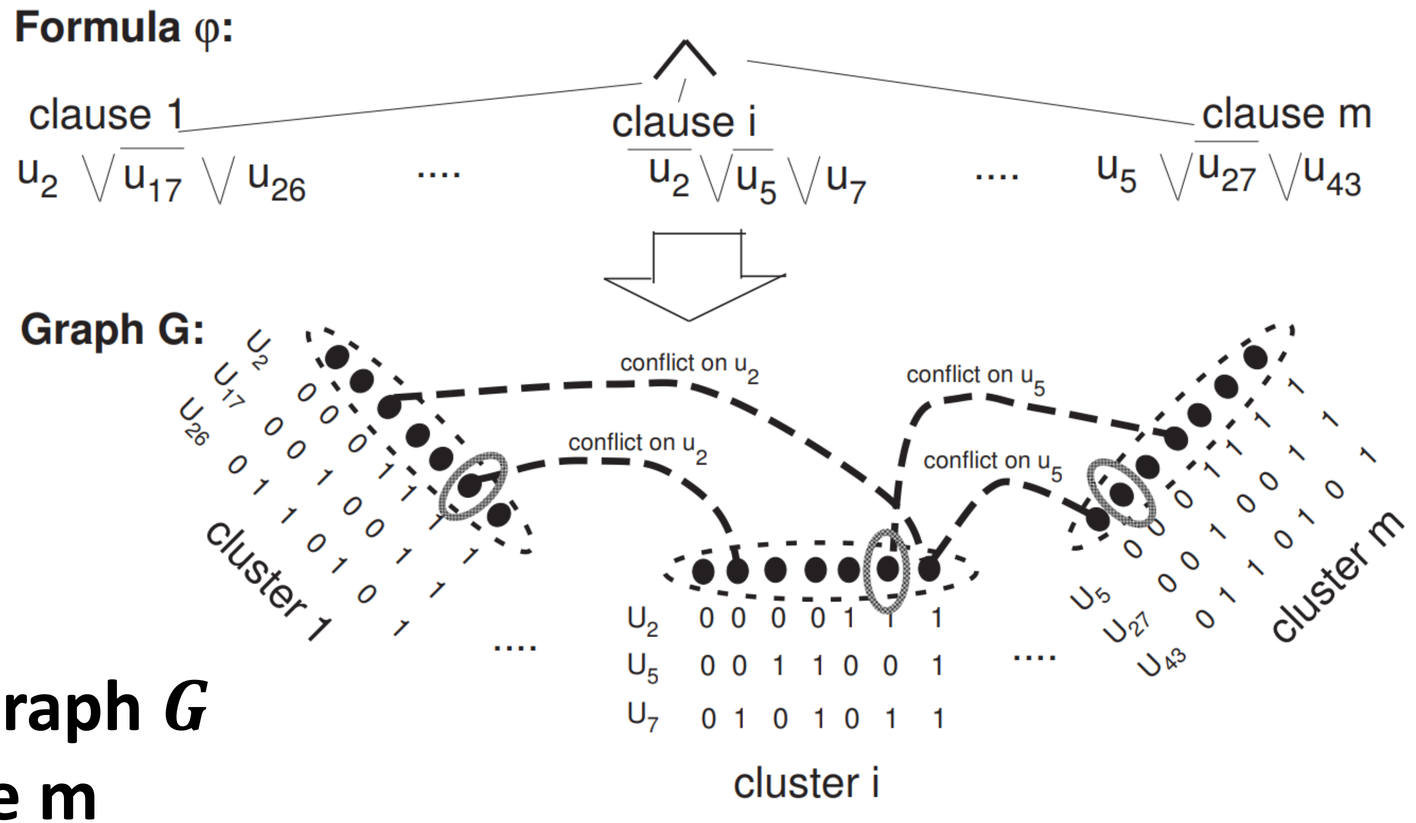
$\overline{U_2} \cup \overline{U_5} \cup U_7$

U_2	0	0	0	0	1	1	1
U_5	0	0	1	1	0	0	1
U_7	0	1	0	1	0	1	1

Missing (1,1,0) assignment since it does not satisfy the clause

ISP is NP-complete

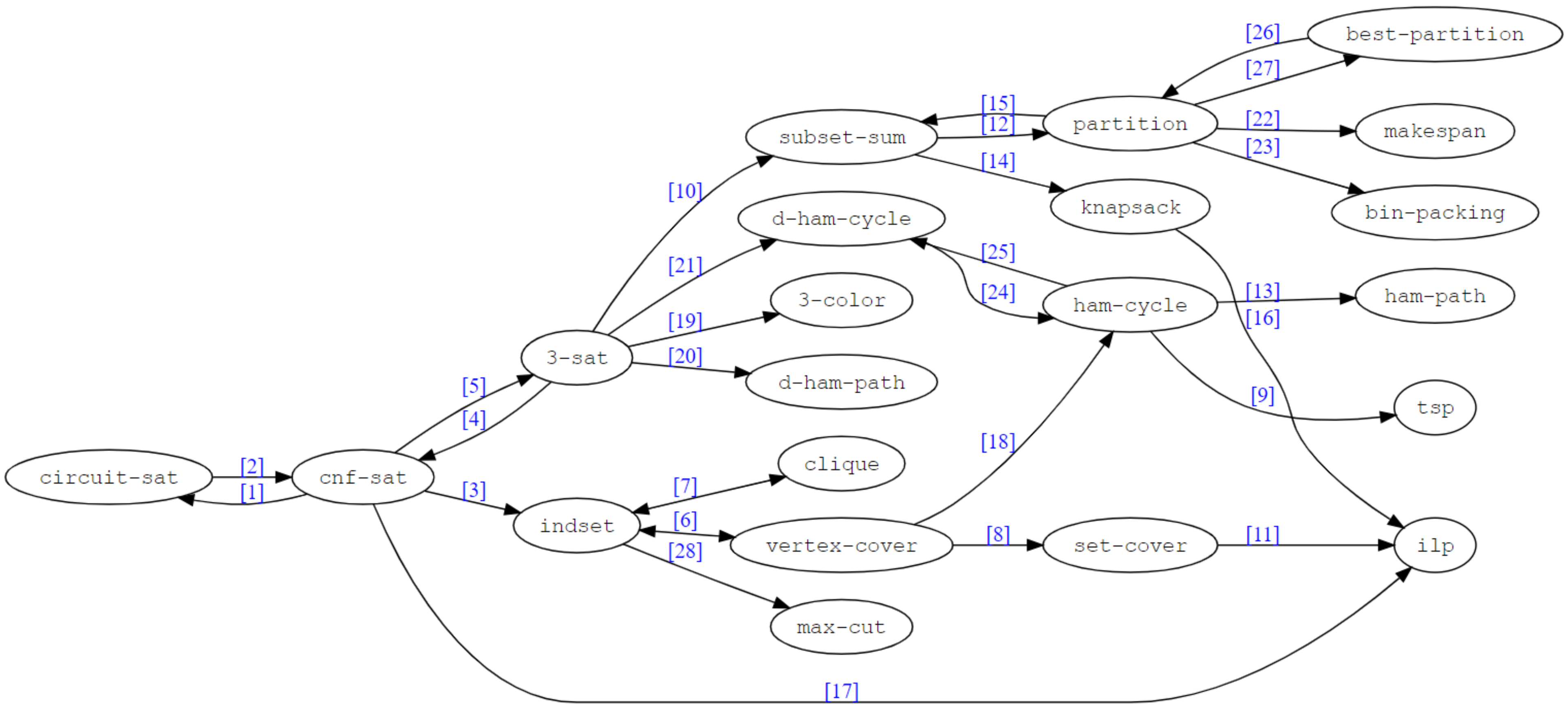
- M-clause φ to 7m-vertex G
- Each cluster describes possib satisfying assignments
- All vertices in a cluster are adjacent



Source: Computational Complexity: A Modern Approach [2]

φ is satisfiable if and only if the graph G has an independent set of size m

Web of reductions



Source: https://sharmaeklavya2.github.io/dl/web-of-reductions/

$\overline{SAT}$

- SAT is NP-complete
- Complement: $A \subseteq X \rightarrow B = \bar{A} = X \backslash \mathrm{A}$

coNP, EXP

- Complement: $A \subseteq X \to B = \bar{A} = X \backslash \mathrm{A}$
- $coNP = \{L\colon \bar{L} \in NP\}$ – has non-empty intersection with NP!
- $\overline{SAT} \in coNP$
- coNP-completeness, ie:
  - $taut = \{\varphi(z)\ is\ satisfied\ by\ any\ \bar{z}\}$
- DTIME: T: $\mathbb{N} \rightarrow \mathbb{N}$, L is in DTIME(T(n)) if $\exists M$ that decides L and runs in $cT(n)$
- $P = \bigcup_{c \geq 1} DTIME(n^c)$
- $(N)EXP = \bigcup_{c \geq 1} D(N)TIME(2^{n^c})$

Mathematical proofs

- Correctness of a proof can be verified by applying a set of axioms to each proof line consequently (which can be polynomial in some axiomatic systems)
- $Theorems = \{(\varphi, 1^n) \colon \varphi\ has\ a\ formal\ proof\ of\ length \leq n\ in\ system\ A\}$ – in NP for any usual system

Discussion

- Result checking is easier than problem solving – creativity as a separation line between complexity classes
- Language of “theorems” is NPC (formal proof of length < smth)
- P = NP? -> automatically create an “easiest” theory for a set of facts (think of Maxwell’s equations for example)
- So, is there something in between NP & NPC?

Ladner’s theorem

- NP-intermediate (NPI) languages: if $P \neq NP$ there exist a language $L \in NP\backslash\mathrm{P}\ so\ that\ L \notin NPcomplete$
- It is unclear if any natural problem is in NPI
- Most known candidates include factoring, minimum circuit size, and graph isomorphism problems (contradictions when assuming some equivalent to P!=NP statements)

Are we doomed if the problem is NP complete?

- NP-completeness means (assuming $P \neq NP$) no polynomial algorithm solves the problem on **every** input
- Fast average time on most common inputs or approximate solutions
- TSP: Euclidian distances + approximation (factor of $1 + \varepsilon$) can yield polynomial algorithm ($n(\log n)^{O(\frac{1}{\varepsilon})}$)

Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems
(https://dl.acm.org/doi/10.1145/290179.290180)

Resources

- ISP NP-completeness - https://www.nitt.edu/home/academics/departments/cse/faculty/kvi/NPC%20INDEPENDENT%20SET-CLIQUE-VERTEX%20COVER.pdf
- Computational Complexity: A Modern Approach (https://theory.cs.princeton.edu/complexity/book.pdf)
- https://www.cs.toronto.edu/~sacook/homepage/1971.pdf
- Introduction to Algorithms, Cormen (i.e. https://web.ist.utl.pt/~fabio.ferreira/material/asa/clrs.pdf)

Backup

More NP problems

- Traveling salesman (TSP)
- Subset sum
- Linear & 0/1 integer programming
- Graph isomorphism
- Composite numbers
- Connectivity
- 1000 more…