1 May 7th, 2019

1.1 Well-Characterized Problems

Definition 1.1. **NP**: Class of decision problems for which there is a "yes-certificate" (i.e. there for yes-instances, there is a "short" proof that the answer is yes)

Note that this is not symmetric, as such, we have the CO-NP class

Definition 1.2. **CO-NP**: Class of decision problems for which there is a "no-certificate"

We know that:

$$\text{Hamiltonian Cycle Problem} \in \text{NP}.$$

However, we still don't know if the Hamiltonian Cycle Problem is in CO-NP. The belief among experts is that:

$$\text{Hamiltonian Cycle Problem} \in \text{CO-NP}.$$

Note that $\text{P} \subset \text{CO-NP}$, since we can just solve the problem for a no-certificate.

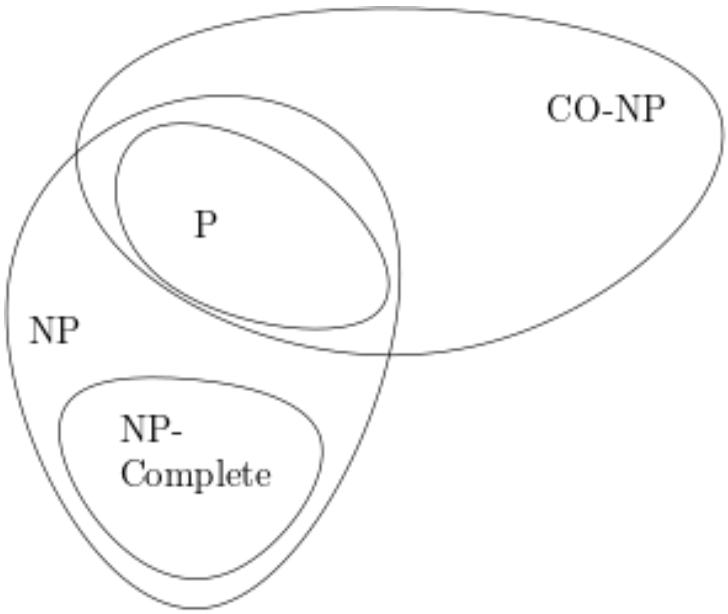

Figure 1: $\text{NP} \cap \text{CO-NP} = \text{P}$?

Definition 1.3. Problem that have both YES and NO certificates, i.e. problem in NP∩CO-NP are said to be **well characterized** or said to have a **good characterization**

Let us consider the decision version of LP: Given a feasible and bounded maximization LP, is there a feasible solution whose value is at least k?

We already know that there is a YES-certificate, as we can just check the feasible solution whose value is at least k, i.e. $\text{LP} \in \text{NP}$.

Remark 1.4 — Note that we did not have to know that there is a poly-time solution for LP, as we are only looking at verifying the YES-certificates

To see whether the LP problem lies in CO-NP, we can consider the dual of the LP. A NO-certificate would be any feasible solution of the dual whose value is less than k, meaning that LP $\in$ CO-NP.

This means that LP is a well-characterized problem. Once again, note that we did not have to use the fact that there is a poly-time algorithm for LP. Historically, when a problem is found to be well-characterized, it is likely that it belongs to P.

Decision Problem for Max Flow: Given a flow network, is there a feasible flow of value greater than or equal to k?

- Feasible flow is a yes-certificate, so this problem belongs to NP.
- The no-certificate is cut, as if there does not exist a flow greater than or equal to k, then there is a cut of value less than k.
- Note that once again we have shown that this problem is well-characterized, without using the knowledge of having a poly-time solution.

Decision Problem for Bipartite Perfect Matching: Given a bipartite graph G with bipartition (A, B) such that $|A| = |B|$, does G have a perfect matching?

- Yes-certificate is just the perfect matching which can be easily checked
- No-certificate: we can use Hall's Theorem, i.e. show that:

$$\exists X \subseteq A, \text{ s.t. } |N(X)| < |X|.$$

  We could also use Konig's Theorem:

$$\exists \text{ vertex cover of size } < n.$$

- As such this problem is well-characterized.

Problems that are well-characterized typically have an associated min-max relationship. These relationships are:

- beautiful and powerful cominatorial results
- lead to poly-time algorithms that are designed around it
- special cases of LP-duality.

We will now look at a well-characterized problem for which we do not know if there is a poly-time algorithm yet - factorization.

1.2 Well-characterization of Factorization

The decision problem for factorization is:

Given two integers x and y, does x have a factor less than y and greater than 1.

To find its first factor, find less than x and greater than 1. If no, then it is prime, if yes, then we can do binary search, but with $y = \frac{x}{2}$. After we find a factor, we can divide it and find the next factor.

Note that it takes $\log x$ calls to find a factor and since 2 is the smallest factor, there can be at most $\log x$ factors ($2 \times 2 \times \ldots \times 2$). As such this will take $\log^2 x$ calls. Since the input is of the order $\log x$, we would be able to check the yes-certificate.

The no-certificate is the prime factorization of x.

1. Check each prime factor and see if it is $\geq y$.
2. We must check if each factor is indeed prime. We can do this with a black box since there is a poly-time algorithm for checking whether a number is prime (Manidra Agarwal).
3. Multiply them together to see if their product is x.
4. After doing this, you should be convinced that the answer is no.

1.3 MAX-SAT Problem

SAT (Satisfiability) was the first problem to be shown was NP-Complete (Proved by Stephen Cook in the early 1970's)
Input:

- n boolean variables (each can be set True or False)
- m clauses: $C_1, C_2, \ldots, C_m$ (each is a disjunction of literals)

Goal: Find an assignment of true/false to the variables that maximizes the number of satisfied clauses.

Remark 1.5 — The original satisfiability problem: Given any boolean formula include "and", "or", "not", can the formula be "satisfied"?

Example 1.6
Let the boolean formula be:

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_4)$$

Can we assign x_i true or false to have the formula be true?

It turns out there this is equivalent (poly-time transformation) from this problem to MAX-SAT.

Definition 1.7. x_1 is a **variable**, and $\overline{x_1}$ is the **negation of a variable**. Collectively, they are **literals**

Definition 1.8. $\vee$: or, **disjunction**
$\wedge$: and, **conjunction**

Definition 1.9. A **clause** is one or more literals combined with disjunctions.

Definition 1.10. A **formula** is one or mor clauses combined with conjunctions

We can rewrite SAT as: Given a set of clauses, is there an assignment of true/false to variable, such that all the clauses are satisfied.

Remark 1.11 — This is clearly in the class NP, as if the answer is yes, we can just give the solution.

MAX-SAT is the maximization version of this problem (note that this is also NP-Hard, as we can easily use this to solve SAT).

1.4 Randomized Algorithm 1

Algorithm

Set each variable x_i to true independently with probability $\frac{1}{2}$.

Analysis

- Let $C_j = x_1 \vee x_2 \vee \ldots \vee x_k$.
- The probability that the clause is satisfied is $1 - \frac{1}{2^k} \geq \frac{1}{2}$.
- The expected number of clauses satisfied is:

$$\geq \frac{1}{2}m \geq \frac{1}{2}OPT.$$

Remark 1.12 — Consider the variable of the problem in which each clause has exactly 3 literals (MAX-3SAT). For this, we have:

$$\text{EXP \# of clauses satisfied} \geq \frac{7}{8}m,$$

meaning that the approximation factor would be $\frac{7}{8}$.

Theorem 1.13
If there is a $\left(\frac{7}{8} + \epsilon\right)$- approximation algorithm for MAX-3SAT for any constant $\epsilon > 0$, then P=NP.

This means that our simple algorithm is the best possible for MAX-3SAT.