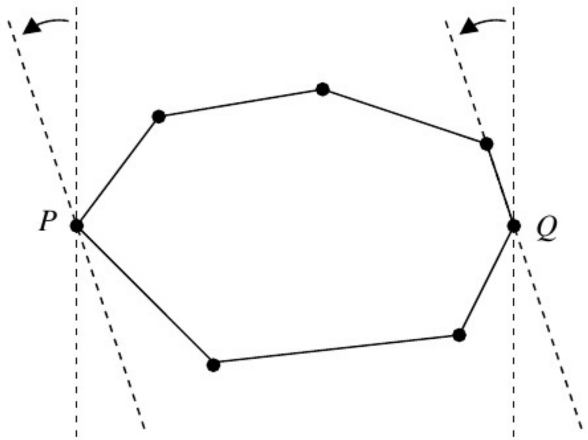
1. [Farthest Pair] Let S be a set of n points in general position in the plane. We want to find P,Q ϵ S such that d(P,Q) is the maximum.

(a) Prove that P and Q are vertices of CH(S).

Let $\text{diam}(S) = d(P,Q)$ with Q not being a vertex of CH(S). By distance-preserving transformations (rotation about the origin and/or reflection about the y-axis), we can assume without loss of generality that PQ is a horizontal line and $x(P) < x(Q)$. Let R be the point with the largest x-coordinate in S. Since Q does not lie on CH(S), we have $x(R) > x(Q)$. Now, it is an easy check that $d(P,R) > d(P,Q)$.

(b) Let Q,Q' be consecutive vertices on CH(S). Let L be the line QQ'. The perpendicular distances of the vertices of CH(S) from L are unimodal. Let P be the farthest point from L. Build a collection C of pairs (P,Q) and (P,Q'). Prove that the farthest pair (P,Q) can be found in C.

Let $\mathrm{diam}(S) = d(P,Q)$. We can assume, without loss of generality, that PQ is horizontal, and $x(P) < x(Q)$. But then, P (resp. Q) is the point in S with the minimum (resp. maximum) x-coordinate. Therefore, $\mathrm{CH}(S)$ lies completely between two vertical lines passing through P and Q. Rotate these two lines simulltaneously (clockwise or counter-clockwise) until one of these lines coincide with an (the first) edge of $\mathrm{CH}(S)$. This procedure is demonstrated in the following figure. It is now an easy check that $(P,Q) \in C$ (in the figure, for example, we have $Q = Q_{i+1}$ and P is a farthest Q_k from the line L_i passing through Q_i and Q_{i+1}).



(c) If the points in S are in general position, what is the maximum size of C?

2n

(d) Propose an $O(n \log n)$-time algorithm for computing the farthest pair in S.

For each edge on CH(S), the vertices most distant from the edge can be obtained by binary search in the unimodal sequence of distances. We do not need to compute the entire sequence, but compute only those distances that are needed during the search.

2. Let Q,Q' be problems such that Q ≤ Q' and Q' ≤ Q. We say that Q and Q' are polynomial-time equivalent. Prove/Disprove: Any two NP-Complete problems are polynomial-time equivalent.

$$Q, Q' \in NPC$$

$$\rightarrow Q, Q' \in NP$$

$$Q \leq Q', \quad Q' \leq Q.$$

What if NP-Complete is replaced by NP-hard?

3. Let G=(V,E) and G'=(V',E') be two undirected graphs. G and G' are called isomorphic if there exists a bijection $f : V \to V'$ such that for all $u,v \in V$, $(u,v) \in E$ if and only if $(f(u),f(v)) \in E'$.

(a) Is GRAPH_ISOMORPHISM ∈ NP? f is the certificate

(b) Is GRAPH_ISOMORPHISM NP-Complete?

"Not known"

Recent result

Quasi-poly-time algo for GI.

4. DOUBLE-SAT: Decide whether a Boolean formula has at least two satisfying assignments. Prove that DOUBLE-SAT is NP-Complete.

Clearly, DOUBLE-SAT is in NP. We reduce SAT to DOUBLE-SAT. Let $\phi = \phi(x_1, \ldots, x_m)$ be an instance for SAT. We convert ϕ to an instance ϕ' for DOUBLE-SAT. If $m = 0$, take $\phi' := \phi \vee y$ for a new variable y. In this case, if $\phi = 0$, then ϕ' has only one satisfying assignment, namely $y = 1$, whereas if $\phi = 1$, then both the choices for y let ϕ' evaluate to 1. So assume that $m \geqslant 1$. Introduce new variables $y_1, \ldots, y_m$ and take $\phi' := \phi(x_1, \ldots, x_m) \vee \phi(y_1, \ldots, y_m)$. If ϕ is unsatisfiable, so is ϕ' too. If ϕ is satisfiable, take a satisfying assignment of $x_1, \ldots, x_m$ for ϕ and let $y_1, \ldots, y_m$ assume any of the $2^m \geqslant 2$ possible values.

5. A CNF formula is called not-all-equal satisfiable if for some truth assignment of the variables, each clause has at least one true literal and at least one false literal.

NAESAT: Decide whether a Boolean formula in CNF is not-all-equal satisfiable.

Prove that NAESAT is NP-Complete.

Clearly, NAE-SAT is in NP. In order to prove its NP-hardness, we use a reduction from CNT-SAT. Let ϕ be an instance for CNF-SAT with m variables $x_1, x_2, \ldots, x_m$. Introduce a single new variable z for the converted formula ϕ'. Let $y_1 \vee y_2 \vee \cdots \vee y_k$ be a clause of ϕ. Convert this to the clause $y_1 \vee y_2 \vee \cdots \vee y_k \vee z$ for ϕ'. I now show that ϕ is satisfiable if and only if ϕ' is NAE satisfiable.

If ϕ is satisfiable by a truth assignment X of $x_1, x_2, \ldots, x_m$, take $z = F$. Then, ϕ' is NAE satisfied for the truth assignment X, F of $x_1, x_2, \ldots, x_m, z$.

Conversely, let ϕ' be NAE satisfied by a truth assignment $Y = (X, b)$ of $x_1, x_2, \ldots, x_m, z$. If $b = F$, then ϕ is satisfied by X. If $b = T$, then ϕ is satisfied by the complement $\overline{X}$ of X.

6. Let P be a problem. The complement problem Q satisfies
Accept(Q) = Reject(P), and Reject(Q) = Accept(P).

Define coNP = {P | The complement of P is in NP}.

Examples of problems in coNP:

✓(a) Complement of SAT
(b) PRIMALITY — a proper divisor
(c) TAUTOLOGY
(d) CONTRADICTION — 0

a Boolean formula that evaluates to 1 for all truth assignments.

Every problem in NP has a succinct certificate.

Every problem in coNP has a succinct disqualification.

7. Prove that if any NP-Complete problem is in coNP, then NP = coNP.

open question

Let $P \in$ coNP be NP-Complete. For every instance I of P, there is a succinct disqualification. Now, take any $Q \in$ NP. Since P is NP-Complete, there is a polynomial-time reduction from Q to P. Convert an instance J for Q to an instance I for P using this reduction. But then, J also has a succinct disqualification, that is, $Q \in$ coNP. It follows that NP $\subseteq$ coNP. A symmetric argument on $\bar{P}$ establishes that coNP $\subseteq$ NP.

8. Prove that $P \subseteq NP \cap coNP$.

P is closed under complement.

$P \subseteq NP$

$P = coP \subseteq coNP$

$P \subseteq NP \cap coNP$.

Open: Is this containment proper?

9. Define coNP-Complete problems.

P is called coNP-complete if

(1) $P \in coNP$

(2) $\forall Q \in coNP, \quad Q \leq P.$

10. Prove that TAUTOLOGY is coNP-Complete.

We propose a reduction from SAT to the complement of TAUTOLOGY. Let ϕ be an input instance for SAT. Convert it to $\sim(\phi)$ (the Boolean complement of ϕ). It is easy to see that ϕ is satisfiable if and only if $\sim(\phi)$ is not a tautology. Therefore the complement of TAUTOLOGY is NP-Hard, that is, TAUTOLOGY is coNP-Hard. Also, TAUTOLOGY is in coNP. It follows that TAUTOLOGY is coNP-Complete.