# NP-Completeness

We are interested in identifying problems in NP which are unlikely to be in P.

A decision problem D is **NP-hard** if for all problems $D' \in NP$ we have $D' \leq D$

Thus NP-hard problems are at least as hard as all NP problems.

**Note that NP-hard problems do not necessarily belong to NP. They could be harder.**

If D is NP-hard and $D \leq D'$ then $D'$ is also NP-hard.

A decision problem D is **NP-complete (NPC)** if

1. $D \in NP$
2. D is NP-hard

NP-complete problems are the hardest problems in NP.

**SAT is NP-complete.**

To see that problem D is NPC show:

1. $D \in NP$ (typically using guess and verify)
2. $D' \leq D$ for some known NPC problem $D'$

Point 2 establishes that D is NP-hard, since $D'$ is NP-hard.

As an example take the Hamiltonian Path Problem (HPP). We have already seen that HPP $\in NP$ by guessing and verifying in p-time. If we can show SAT $\leq$ HPP then we can conclude that HPP is NPC. It is indeed possible to show SAT $\leq$ HPP.

So HPP is NP-complete.

# Intractability via NP-completeness

Our original interest was in deciding which problems are tractable and which are intractable. We have a definition for ‘tractable’, and we have confirmed that problems such as sorting are indeed tractable.

What we have not done is show that a problem is intractable.

We now see how NP-completeness can help with this.

**Suppose $P \neq NP$. If a problem D is NP-hard then $D \notin P$**

**Proof.** Assume P $\neq$ NP and $D$ is NP-hard.

Suppose for a contradiction that $D \in$ P. We show that NP $\subseteq$ P. Take $D' \in$ NP. Since $D$ is NP-hard, we have $D' \leq D$. Hence $D' \in$ P. We have shown that NP $\subseteq$ P.

But we know P $\subseteq$ NP. Hence P $=$ NP which contradicts our assumption.

Thus if we can show that a problem is NPC, we know that it is intractable (assuming that $P \neq NP$).

Recall the Travelling Salesman Problem TSP: **Given a (complete) weighted graph $(G, W)$, find a tour of G of minimum weight which visits each node exactly once and returns to the start node**

TSP is an optimisation problem. We first define a decision version of TSP which we call $TSP(D)$: Given a weighted graph $(G, W)$ and a bound B, is there a tour of G with total weight $\leq$ B?

We can think of B is being a budget or travel allowance. In the decision version, we ask whether there is a tour that does not exceed the budget.

We show that TSP(D) is NP-complete using Method 3.4.4:

1. TSP(D) $\in$ NP:

   If we guess a path $p$, we can check in p-time that $p$ is a Hamiltonian circuit of $G$ and that $W(p) \leq B$. Clearly $|p| \leq |G|$.

   More formally define $\text{V}\small{\text{ER}}$-TSP(D)$((G,W),B,p)$ iff $p$ is a Hamiltonian circuit of $(G,W)$ and $W(p) \leq B$.

   Then TSP(D)$((G,W),B)$ iff $\exists p.$VER-TSP(D)$((G,W),B,p)$.

   Also if VER-TSP(D)$((G,W),B,p)$ then $|p| \leq |G|$ under reasonable definitions of size.

2. $D' \leq \mathrm{TSP(D)}$ for some known NPC problem $D'$:

   We choose HAMPATH as the known NPC problem and show HAMPATH $\leq \mathrm{TSP(D)}$.

   We need to define a p-time function $f$ which transforms a graph $G$ into a weighted graph $(G', W)$ together with a bound $B$ so that HAMPATH$(G)$ iff $\mathrm{TSP(D)}((G', W), B)$.

   Given $G$ we construct $(G', W)$ as follows: Set nodes$(G')$ = nodes$(G)$. Given any two distinct nodes $x, y$ of $G$:

   - If $(x, y)$ is an arc of $G$ then $(x, y)$ is also an arc of $G'$, with $W(x, y) = 1$.
   - If $(x, y)$ is not an arc of $G$ then $(x, y)$ is an arc of $G'$, with $W(x, y) = 2$.

   Thus we add in the missing arcs of $G$ but we give them a higher weight. Finally we let $B = n + 1$ where $G$ has $n$ nodes.

   It is not hard to see that $f(G) = ((G', W), B)$ is p-time — easiest to see using the adjacency matrix representation.

   We now check that $f$ is a reduction: HAMPATH$(G)$ iff $\mathrm{TSP(D)}((G', W), B)$: Suppose $G$ has a Hamiltonian path $\pi$ with endpoints $x$ and $y$. The same path in $G'$ has weight $n - 1$ (all arcs have weight 1). We get a Travelling Salesman tour by adding in arc $(x, y)$ with $W(x, y) \leq 2$. Thus we have a tour of weight $\leq n + 1 = B$.

   Conversely, suppose $(G', W)$ has a tour of weight $\leq B = n + 1$. This has $n$ arcs. So at most one arc can have weight 2. Suppose this arc has endpoints $x, y$. Then omitting arc $(x, y)$ gives us a Hamiltonian path in $G$.

We conclude that $TSP(D)$ is NP-complete. Finally we can show that TSP is intractable (assuming $P \neq NP$).

- Suppose that TSP can be solved by a p-time algorithm
- We compute the optimal value O in p-time
- Then we can also solve TSP(D) in p-time — we simply check whether $O \leq B$
- So $TSP(D) \in P$
- But this is impossible since TSP(D) is NP-complete and we assume $P \neq NP$

Check slides for 2 more examples