NP - Hardness

Not all problems are solvable with a reasonable efficiency. This categorization aims to partition problems based upon how "efficiently" solvable they are.

An example of a problem which is not solvable would be the coloring problem of graphs or a modified scheduling problem.

Given jobs $j_1, \dots, j_n$ and duration $d_1, \dots, d_n$ respectively; what is the optimal scheduling of these jobs on k identical processors, $P_1, \dots, P_k$?

No efficient algo found till date

Class P

A problem π belongs to the class P if there exists an algorithm which finds the correct result for any input x in time $O(\text{poly } |x|)$.

Class NP

A problem π belongs to NP if there exists a polynomial time verifying algorithm T and a possible polynomial sized assignment y for the case given by x such that:

- If x is a positive assignment; $\exists y.\, T(x,y) = \text{true}$
- If x is a negative assignment; $\forall y.\, T(x,y) = \text{false}$

Note that $P \subset NP$.

In the above definition, π is a deterministic problem. That is, instead of "Find valid coloring," we ask "Does there exist a coloring?" Similarly, "Find $\boxed{\text{shortest}}$ path" becomes "Does there exist a path of length $\boxed{\text{atmost}}$ v?" where v is input.

Polynomial Reduction

Consider two problems π_1 and π_2. We say that π_1 is polynomial time reducible to π_2 if there exists a polynomial time function f such that for every input w, $w \in \pi_1 \leftrightarrow f(w) \in \pi_2$.

We are essentially stating that π_2 is atleast as hard as π_1; and that we can solve π_2 if we can solve π_1.

π_1 is polynomial time reducible to $\pi_2 \Rightarrow$ $\pi_1 \leq_m \pi_2$

NP-Hard

A problem π is NP-hard if $\pi \in NP$ and for every problem $\pi' \in NP$,

$$\pi' \leq_m \pi$$

A problem which both belongs to NP and is NP-hard is said to be NP-complete.

Since many important problems are NP-hard in nature, we try to give heuristics which work good enough for practical situations. For optimization problems, we can try to give C-approximate algorithms which differ from the optimal by a factor of c.