NP-Completeness

Algorithmic Approaches to NP-Complete Problems

Algorithms: Design and Analysis, Part II

NP-Completeness: The Beginning, Not the End

Question: So your problem is NP-complete. Now what?

Important: NP-completeness not a death sentence.

⇒ but, need appropriate expectations / strategy

Three Useful Strategies

① focus on computationally tractable special cases

Examples: - WIS in path graphs (and trees, bounded treewidth) (NP-C in general graphs)

- Knapsack with polynomial size capacity (e.g., $W = O(n)$)
- 2SAT (P) instead of 3SAT (NP-C)
- vertex cover when OPT is small

Three Useful Strategies (con'd)

(2) heuristics – fast algorithms that are not always correct

Examples (forthcoming): greedy and dynamic programming-based heuristics for knapsack.

(3) solve in exponential time but faster than brute-force search

- knapsack ($O(nW)$ instead of 2^n)
- TSP ($\approx 2^n$ instead of $\approx n!$)
- Vertex Cover ($\approx 2^{OPT} n$ instead of n^{OPT})

} forthcoming