TRACTABLE AND INTRACTABLE PROBLEMS

Limits of Computation

MARCH 3, 2020

DR. JAD MATTA

Tractable and Intractable problems

Tractable problems: the class P

The class of algorithms whose time-demand was described by a polynomial function. Such problems are said to be tractable and in the class PTIME.

A problem is in P if it admits an algorithm with worst-case time-demand in O (n^k) for some integer k.

However there are some problems for which it is known that there are no algorithms which can solve them in polynomial time, these are referred to as provably intractable and as being in the class **EXPTIME (exponential time)** or worse. For these problems, it has been shown that the lower bound on the time-demand of any possible algorithm to solve them is a function that grows unreasonably fast.

Intractable problems: the class EXPTIME and beyond

A problem is in the class EXPTIME if all algorithms to solve it have a worst-case time demand which is in O (2^p(n)) for some polynomial p(n).

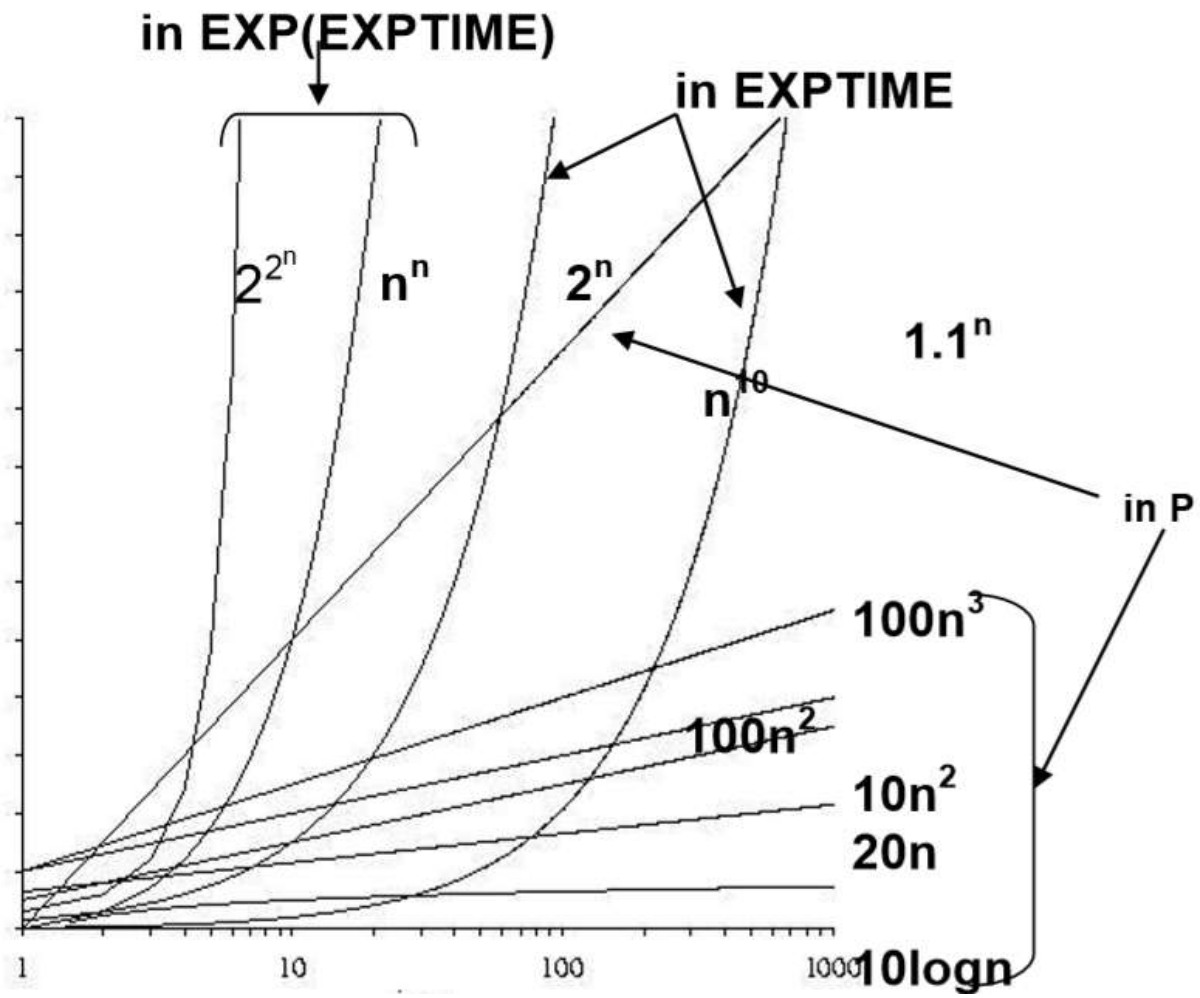
Higher time-complexity classes

There are other classes of problems for which the time demand cannot be bounded above even by a function of the form 2^p(n).

In fact there are a hierarchy of these higher time-complexity classes such that a problem within a given class is considered more intractable than all those within lower-ranked classes.

So beyond EXPTIME we can have EXP(EXPTIME), for which the time-demands of all known solutions are bounded above by a multiple of 2 ^ 2 ^ p(n).

All these classes of provably intractable problems, from EXPTIME upward, can be referred to as having a super-polynomial time demand.



It turns out that the most interesting class of problems is a class, which lies in some sense between the class of tractable problems P and those of the provably intractable, super-polynomial time problems.

The classes NP and NP-Complete

The Hamiltonian Circuit Problem

A connected, undirected, unweighted graph G has a Hamiltonian circuit if there is a way to link all of the nodes via a closed route that visits each node once, and only once.

As the size grows the time-demand appears to scale very badly and it is strongly believed that there are no polynomial time algorithms for this problem.

The Travelling Salesman Problem

The TSP shares the extremely bad scaling behavior as the Hamiltonian circuit problem, and is one of the famous intractable problems. This graph problem is similar to the Hamiltonian cycle problem in that it looks for a route with the same properties as required by the Hamiltonian cycle, but now of minimal length.

The Hamiltonian cycle and Travelling Salesman Problems belong to the class of NP-Complete, which is a subset of the larger problem class NP. NP-Complete is a class of problems whose time-complexity is presently unknown, though strongly believed to be super-polynomial.

Polynomial time (p-time) reduction

A problem A reduces in =time to another problem B, written as

$$A \leq_p B$$

Means that there is some procedure, taking no more than polynomial time as a function of the size of the input to A, which

- Converts an input instance of A into an input instance of B
- Allows a suitable algorithm for problem B to be executed
- Provides a mechanism whereby the output obtained by this algorithm for problem B can be translated back into an output for problem A

The algorithm for problem B thus also provides a solution to problem A. Moreover A's solution will be obtained in a time which is in the same complexity class as the algorithm which solves B, since the extra work needed to translate is just in p-time.

Example: To show that Hamiltonian cycle is reducible to travelling salesman decision problem.

- Take an instance of Hamiltonian Cycle problem, say G.
- Create a new weighted graph (G', w) as follows:
    - Node of G' are the same as nodes of G
    - Add extra edges so that G' is fully connected (so that it now has n/2 (n-1) edges

    - Set the weights in the new graph G' so that if an edge existed already in G it has weight 0, otherwise it has a weight 1
- Return Travelling salesman Decision Problem ((G', w), 0)

The reduction takes time $O(n^2)$ in the number of nodes since the maximum number of edges in any undirected graph is only n/2 * (n -1), and this the number of added edges must be bounded above this.

There are three basic defining properties of problems in NP and NP-complete.

1. ***Problems in NP and NP-Complete are very hard to solve but easy to check***

The problems are hard because they appear to admit algorithms whose time-demand behavior is exponential. However if a solution to a yes-instance of the problem is asserted then it can be checked in polynomial time; this ability to check a solution for correctness in polynomial time is referred to as a short certificate for the problem.

The class of problems with this "very hard to solve, but easy to check" property is known as Non-deterministic Polynomial.

2. ***Problems in NP-Complete are the hardest problems in NP***

An NP-Hard problem is one to which any problem in NP can be reduced in polynomial time:

If A is NP-hard, for all B in NP it is true that B $\leq_p$ A.

The class NP-Complete is the class of problems within NP itself, which have this property:

$$\textbf{NP-Complete} = \textbf{NP} \cap \textbf{NP-hard}$$

3. ***Problems in NP-Complete stand or fall together***

Any problem in the class NP-Complete can be shown to be reducible in polynomial time to any other problem in the class. Meaning that there is a way in which any problem A can be mapped onto any other problem B using a number of steps taking no more than

polynomial time such that a solution for B also provides a solution for A, and that the converse can also be done.

If A, B Ɛ NP-Complete then A $\leq_p$ B (A reduces in p–time to B) and B $\leq_p$ A (B reduces in p-time to A)

It is the best-known property of problems in NP-Complete because it means that should a polynomial-time algorithm be found for just one problem in NP-Complete, then all NP-Complete problems would be solvable in polynomial time.

How to assign a new problem to NP-Complete?

It would need to be shown that:

A $\leq_p$ B1 and B2 $\leq_p$ A, for some B1, B2 in NP-Complete