The P versus NP question: is there any progress?

Overview

Recall the P versus NP question.
Touch upon its significance.
What we have learned in the past 30 years.

Too broad a subject to cover in depth.

Efficient Computation

Example: multiplying two n-digit numbers, A and B.
Exponential time algorithm: add A to itself B times.
Polynomial time algorithm: as in school. Time O(n^2).
Multiplication is in P (has a polynomial time algorithm).

Factoring: Given an n-digit number, find its prime factors. No known polynomial time algorithm (though there are sub-exponential algorithms).
Factoring is not known to be in P (but it might be).

Why Polynomial Time?

Requires understanding (unlike exhaustive search).

Moderate growth of time complexity in terms of problem size.

Robust notion across different computer architectures, problem encodings.

Often, agrees with practical efficiency.

NP
Nondeterministic Polynomial Time

Problems for which finding a solution might be difficult, but verifying correctness is easy.

Many examples: factoring, Soduko puzzles*, rearranging Rubic's cube*, scheduling subject to constraints, finding a short traveling salesperson tour, coloring a planar map with 4 colors …

The above problems are in NP (have a nondeterministic polynomial time algorithm).

Is P=NP?

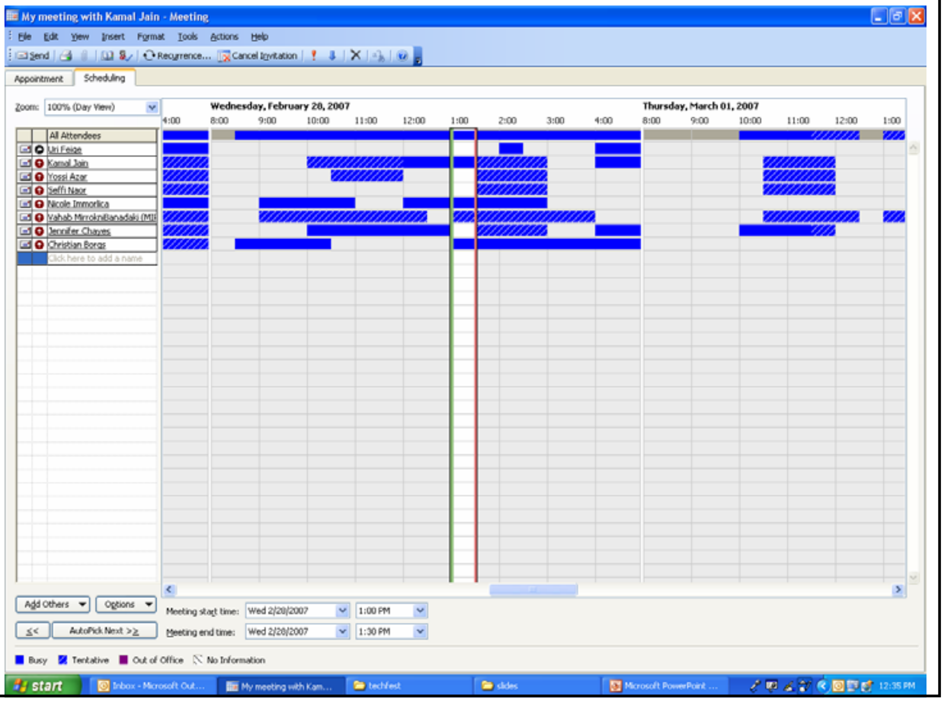
Is there a systematic search procedure (algorithm) that would efficiently solve any well defined computational problem?

Reductions Among Computational Problems

Essentially: an instance of problem A can be recast as an instance of problem B.

Solving problem A does not require more computational resources than problem B (up to polynomial factors).

Simple example: scheduling meetings reduces to graph coloring and vice versa.


My meeting with Kamal Jain - Meeting
File Edit View Insert Format Tools Actions Help
Send Recurrence... Cancel Invitation
Appointment Scheduling
Wednesday, February 28, 2007
Thursday, March 01, 2007
4:00 8:00 9:00 10:00 11:00 12:00 1:00 2:00 3:00 4:00 8:00 9:00 10:00 11:00 12:00 1:00
Add Others Options
<< AutoPick Next >>
Busy Tentative Out of Office No Information
start Inbox - Microsoft Out... My meeting with Kam... techfest slides Microsoft PowerPoint ... 12:35 PM

Zoom: 100% (Day View)

All Attendees
Uri Feige
Kamal Jain
Yossi Azar
Seffi Naor
Nicole Immorlica
Vahab Mirrokni Banadaki (MI
Jennifer Chayes
Christian Borgs
Click here to add a name

Meeting start time: Wed 2/28/2007 1:00 PM
Meeting end time: Wed 2/28/2007 1:30 PM

A Simplified Scheduling Problem

All people initially free on Wednesday.

Multiple meeting requests: each including a list of required attendees. A meeting takes one hour.

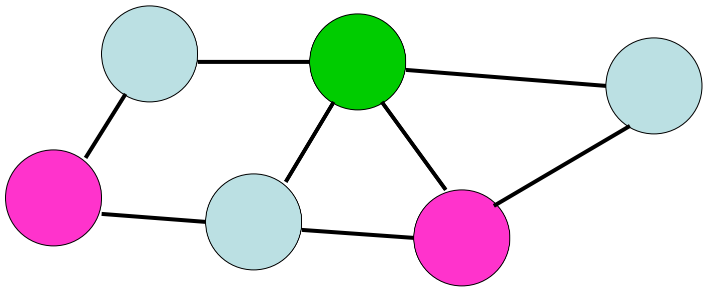
Constraint: Two meetings can be scheduled at the same hour only if no person is required to attend both meetings.

Goal: schedule all meetings in an 8 hour period.

Graph Coloring

Input: a graph and a parameter k.

Goal: a legal vertex coloring using at most k colors.



The Simple Reduction

View each meeting as a vertex.

View each conflict (two meetings sharing the same attendee) as an edge.

View each hour as a color class. k=8.

A legal k-coloring is a feasible schedule.

An algorithm for solving the coloring problem would then also solve the scheduling problem.

Not all reduction are so simple.

Cook-Levin Theorem

Every problem in NP reduces to SAT (satisfiability of Boolean formulas, a problem motivated by formal logic).

In a sense, SAT is at least as hard as any problem in NP. It is NP-hard.

Being also in NP, it is NP-complete.

P=NP is the same question as whether SAT is in P.

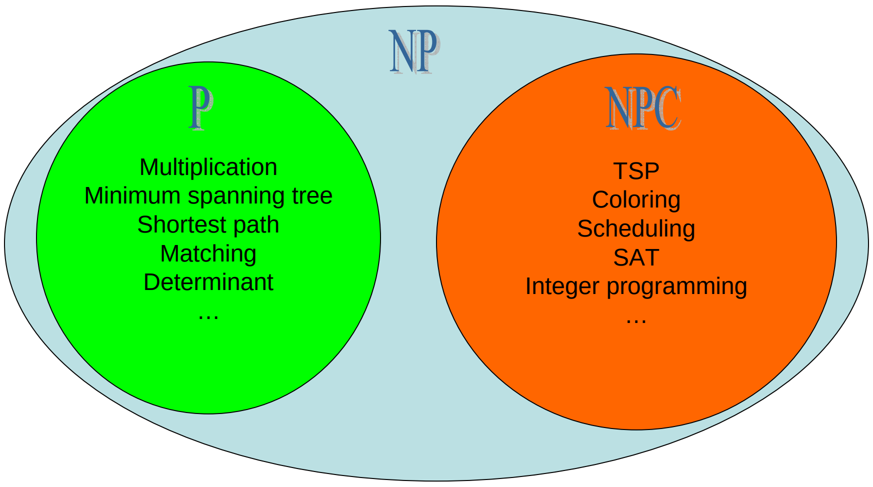
The Rise of NP-completeness

Karp: SAT is no exception – many optimization problems in operations research are NP-complete (coloring, Travelling Salesperson TSP, scheduling, and dozens more).

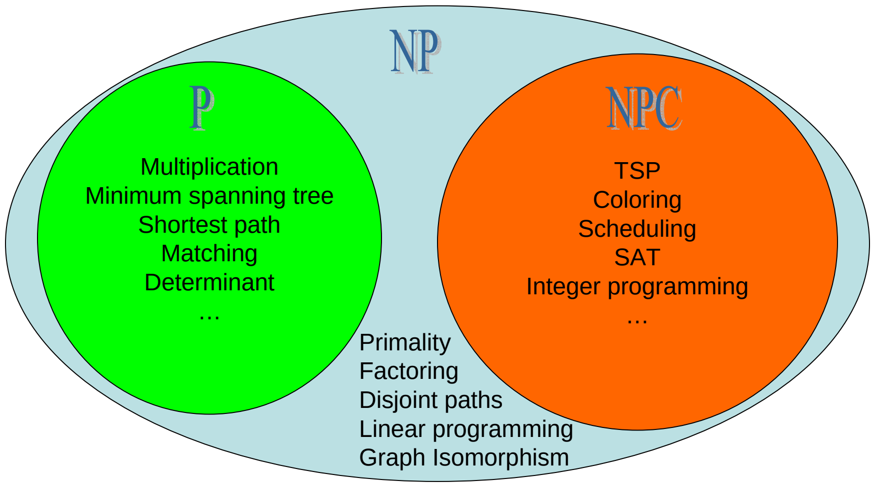
In fact, almost all problems in NP appear to be either easy (in P) or “provably” difficult (NP-complete).

Mid 70ies - book by Garey and Johnson.

The World in the Mid 70ies



The World in the Mid 70ies



Significance of P versus NP

Significance extends well beyond computer science. Briefly Touch upon significance in

- Mathematics
- Philosophy
- Biology
- Economics

Mathematics

If P=NP, then finding a proof is not much more difficult that verifying it.

Can theorem proving be automated?

(If yes, then does the world need mathematicians?)

Philosophy

Seeing a simple solution to a problem that you thought was difficult, did you ever ask yourself “why didn’t I think of that?”

Were you disappointed that you didn’t?

You shouldn’t be!

If there is a systematic and efficient way of finding the right answers (those that sound right), then P=NP…

Biology

Common biological wisdom: proteins fold to the minimum energy configuration.

Finding the minimum energy 3-dimensional configuration of a protein is NP-hard.

Perhaps proteins do not always fold to the minimum energy configuration?

Economics

Do economic theories of free markets (e.g., market prices reach an equilibrium) make sense?

Many notions of equilibrium used in game theory and economics (e.g., Nash Equilibrium) are believed not to be in P.

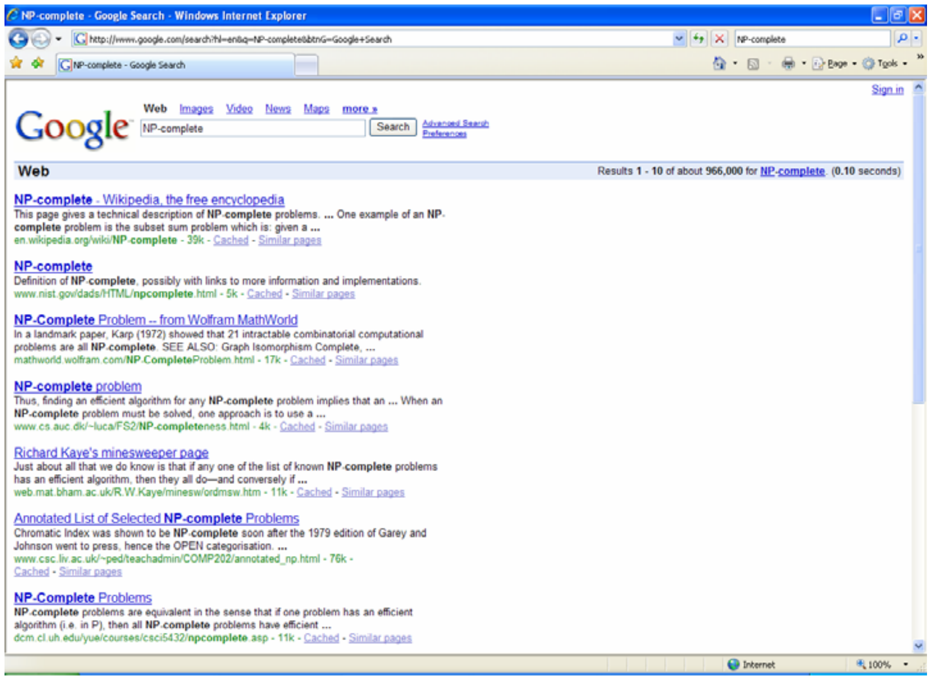
Hence how can the market “compute” them?

Stature of NP-completeness

Extensively studied in computer science and referenced in other sciences.

Hundreds of (good) scientific papers every year.

Clay Mathematics Institute listed P versus NP as one of the 7 millennium problems. $1000000 prize per problem.



NP-complete - Google Search - Windows Internet Explorer

http://www.google.com/search?hl=en&q=NP-complete&btnG=Google+Search

Web Images Video News Maps more »

NP-complete

Web — Results **1** - **10** of about **966,000** for NP-complete. (**0.10** seconds)

NP-complete - Wikipedia, the free encyclopedia
This page gives a technical description of **NP-complete** problems. ... One example of an **NP-complete** problem is the subset sum problem which is: given a ...
en.wikipedia.org/wiki/**NP-complete** - 39k - Cached - Similar pages

NP-complete
Definition of **NP-complete**, possibly with links to more information and implementations.
www.nist.gov/dads/HTML/**npcomplete**.html - 5k - Cached - Similar pages

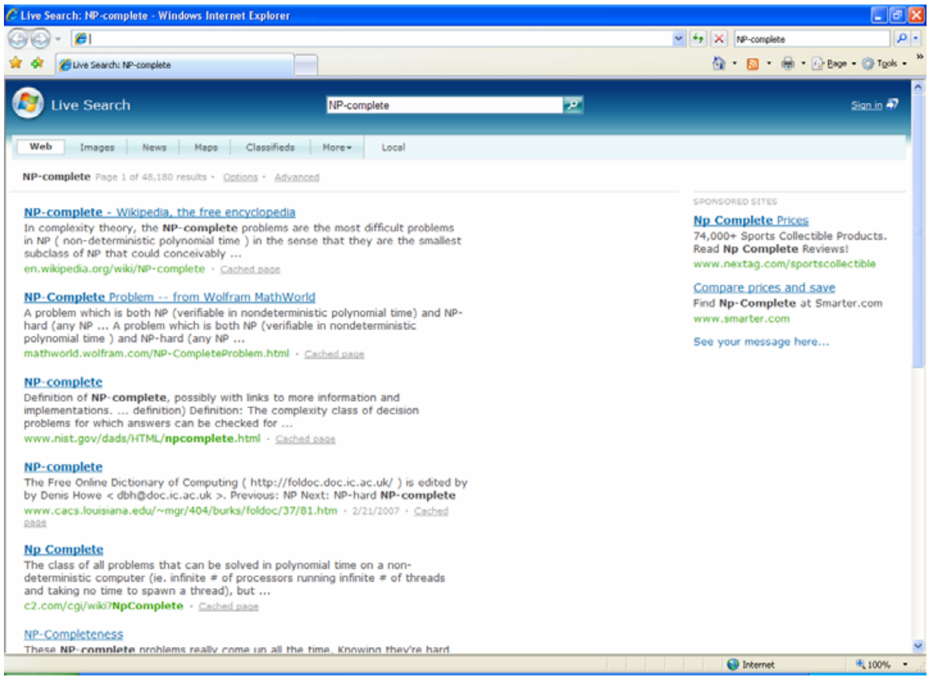
NP-Complete Problem -- from Wolfram MathWorld
In a landmark paper, Karp (1972) showed that 21 intractable combinatorial computational problems are all **NP-complete**. SEE ALSO: Graph Isomorphism Complete, ...
mathworld.wolfram.com/**NP-Complete**Problem.html - 17k - Cached - Similar pages

NP-complete problem
Thus, finding an efficient algorithm for any **NP-complete** problem implies that an ... When an **NP-complete** problem must be solved, one approach is to use a ...
www.cs.auc.dk/~luca/FS2/**NP-complete**ness.html - 4k - Cached - Similar pages

Richard Kaye's minesweeper page
Just about all that we do know is that if any one of the list of known **NP-complete** problems has an efficient algorithm, then they all do—and conversely if ...
web.mat.bham.ac.uk/R.W.Kaye/minesw/ordmsw.htm - 11k - Cached - Similar pages

Annotated List of Selected **NP-complete** Problems
Chromatic Index was shown to be **NP-complete** soon after the 1979 edition of Garey and Johnson went to press, hence the OPEN categorisation. ...
www.csc.liv.ac.uk/~ped/teachadmin/COMP202/annotated_np.html - 76k - Cached - Similar pages

NP-Complete Problems
NP-complete problems are equivalent in the sense that if one problem has an efficient algorithm (i.e. in P), then all **NP-complete** problems have efficient ...
dcm.cl.uh.edu/yue/courses/csci5432/**npcomplete**.asp - 11k - Cached - Similar pages



NP-complete Page 1 of 48,180 results · Options · Advanced

NP-complete - Wikipedia, the free encyclopedia
In complexity theory, the **NP-complete** problems are the most difficult problems in NP (non-deterministic polynomial time) in the sense that they are the smallest subclass of NP that could conceivably ...
en.wikipedia.org/wiki/NP-complete · Cached page

NP-Complete Problem -- from Wolfram MathWorld
A problem which is both NP (verifiable in nondeterministic polynomial time) and NP-hard (any NP ... A problem which is both NP (verifiable in nondeterministic polynomial time) and NP-hard (any NP ...
mathworld.wolfram.com/NP-CompleteProblem.html · Cached page

NP-complete
Definition of **NP-complete**, possibly with links to more information and implementations. ... definition) Definition: The complexity class of decision problems for which answers can be checked for ...
www.nist.gov/dads/HTML/**npcomplete**.html · Cached page

NP-complete
The Free Online Dictionary of Computing (http://foldoc.doc.ic.ac.uk/) is edited by by Denis Howe < dbh@doc.ic.ac.uk >. Previous: NP Next: NP-hard **NP-complete**
www.cacs.louisiana.edu/~mgr/404/burks/foldoc/37/81.htm · 2/21/2007 · Cached page

Np Complete
The class of all problems that can be solved in polynomial time on a non-deterministic computer (ie. infinite # of processors running infinite # of threads and taking no time to spawn a thread), but ...
c2.com/cgi/wiki?**NpComplete** · Cached page

NP-Completeness
These **NP-complete** problems really come up all the time. Knowing they're hard

SPONSORED SITES

Np Complete Prices
74,000+ Sports Collectible Products. Read **Np Complete** Reviews!
www.nextag.com/sportscollectible

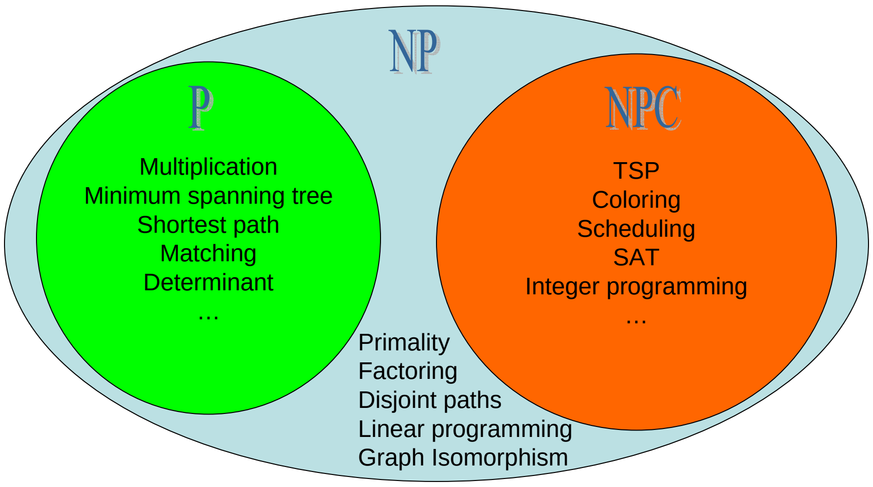
Compare prices and save
Find **Np-Complete** at Smarter.com
www.smarter.com

See your message here...

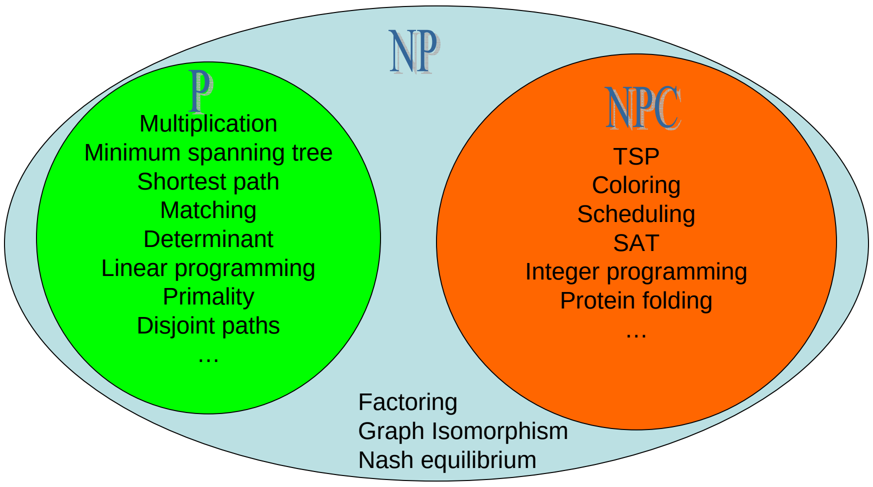
What We Know Now that We Did Not Know Then

- More problems classified in P and NPC.
- Attempts at resolving the P versus NP problems failed.
- Modern cryptography.
- Randomized algorithms.
- Quantum computing.
- Approximation algorithms and hardness of approximation. The PCP theorem.

The World in the Mid 70ies



The World Today



Important Problems Moved Into P

Linear programming [Khachiyan, Karmarkar]: a very strong general purpose algorithmic framework.

Disjoint paths [Robertson and Seymour]: as part of the deep and beautiful theory of graph minors.

Primality [Agrawal, Saxena, Kayal]: a very basic question in mathematics. (Even showing that it is in NP requires some number theory.)

The Main Remaining Offenders

Factoring. Not to be confused with primality. Of key importance in cryptography (RSA).

Graph isomorphism.

Nash equilibrium and similar fix-point notions from game theory and economics. In contrast, the von-Neuman minimax value for two person 0-sum games is in P (using linear programming).

Trying to Prove P=NP

Suffices to show that one NP-hard problem has a polynomial time algorithm.

Many such attempts. For example, try to formulate TSP as a polynomial size linear program.

No success.

Moreover, proofs that some specific approaches must fail.

Proving Lower Bounds

To prove that P differs from NP, show that any algorithm for SAT must require superpolynomial time on some instances.

Essentially no progress in proving lower bounds (except for specialized models).

But we understand better why we fail (large classes of approaches cannot work – relativization, natural proofs).

Computational Difficulty as a Blessing

Cryptography (encryption, digital signatures, …).

Breaking a crypto-system better not be in P, but should be in NP (guess the secret key).

Lives in the area between P and NPC.

Clear why not in P.

More difficult to explain why not NPC.

Randomness

- Invades many areas of computation.
- Modern cryptography must use randomness. If there is no randomness, there are no secrets.
- Many heuristics (general purpose algorithms that are not guaranteed to always work) employ random steps or random initialization: genetic algorithms, simulated annealing, belief propagation, etc.
- On huge data sets, many decisions are based on random samples.

Randomized Algorithms

Use randomness so as to provably have high success probability on every input (unlike heuristics that succeed on some inputs and fail on others).

For every input, on most random choices of the algorithm, it outputs the right answer.

- Testing polynomial identities.
- Approximating the volume of a convex high dimensional body.

Quantum Computing

Theoretical probabilistic computation model based on quantum physics.

Allows quantum parallel hardware to have exponential effect in some special cases (unlike standard parallelism).

NPC problems are believed not to have quantum poly-time algorithms.

Factoring has quantum poly-time algs [Shor].

Motivates building quantum computers, or revising quantum mechanics.

Microsoft Station Q (Mike Freedman).

Approximation Algorithms

A method of handling NP-hard optimization problems.

Trade optimality against efficiency.

Using 1% of the effort, recover 99% of the value.

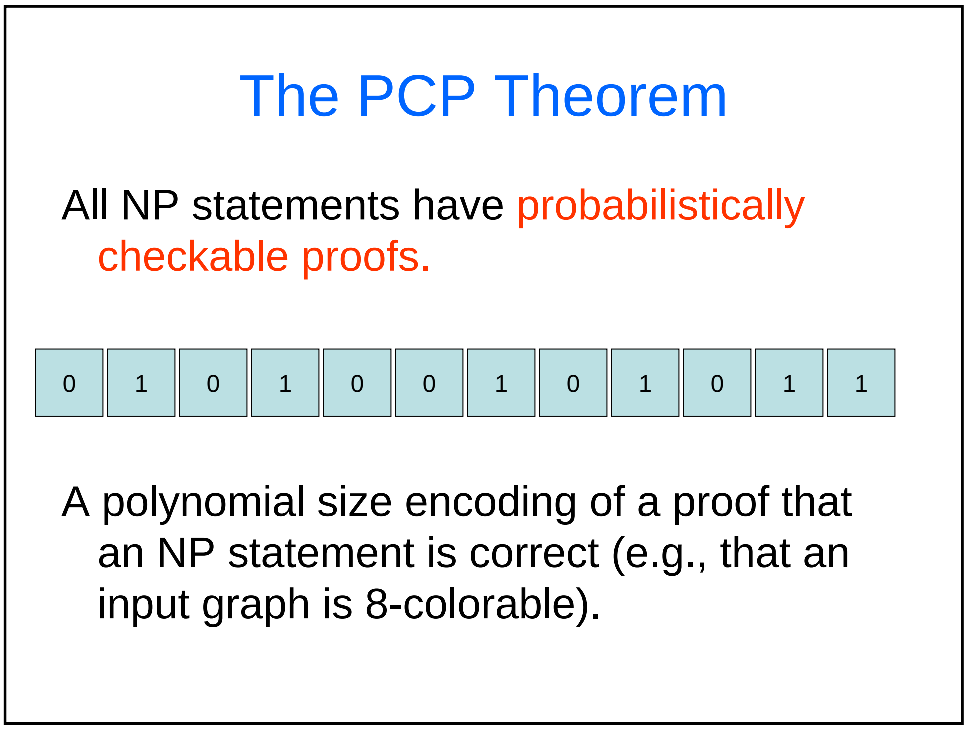
Example

Finding shortest TSP tour (a tour that visits all cities) is NP-hard.

Is there an algorithm that efficiently finds a near optimal tour?

To some extent, yes – in polynomial time one can find a tour of length no more than 3/2 times the optimum. (Approximation ratio 3/2.)

Can one do even better?

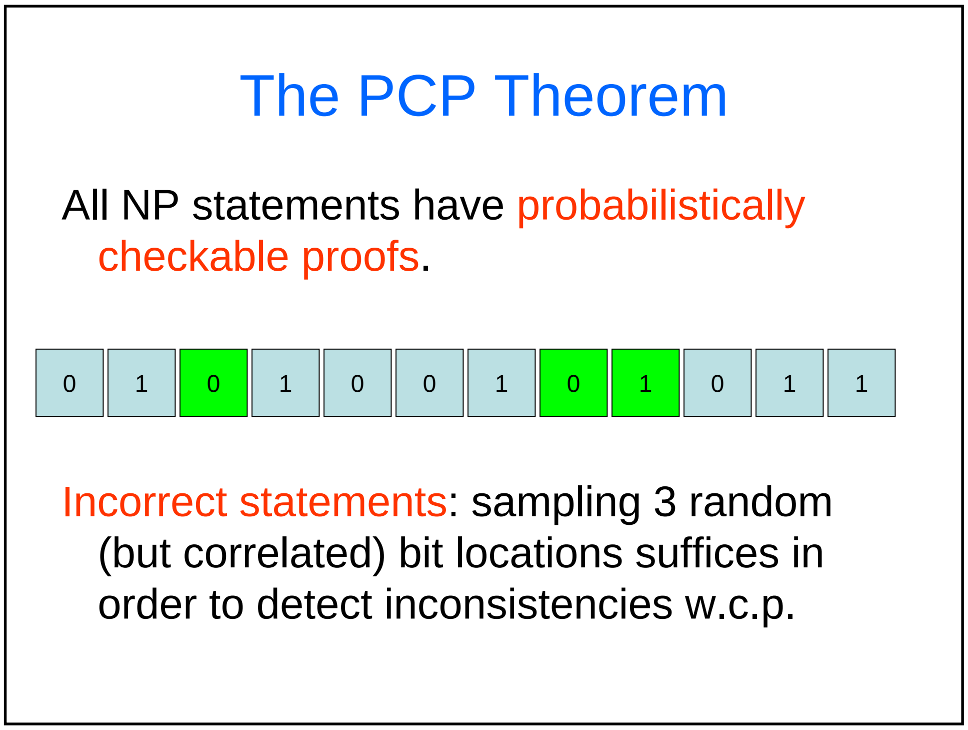


The PCP Theorem

All NP statements have probabilistically checkable proofs.

0	1	0	1	0	0	1	0	1	0	1	1

A polynomial size encoding of a proof that an NP statement is correct (e.g., that an input graph is 8-colorable).



The PCP Theorem

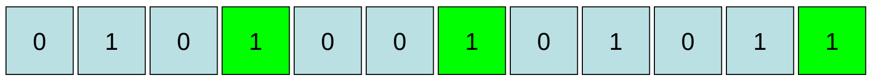
All NP statements have probabilistically checkable proofs.

0	1	0	1	0	0	1	0	1	0	1	1

Incorrect statements: sampling 3 random (but correlated) bit locations suffices in order to detect inconsistencies w.c.p.

The PCP Theorem

All NP statements have probabilistically checkable proofs.



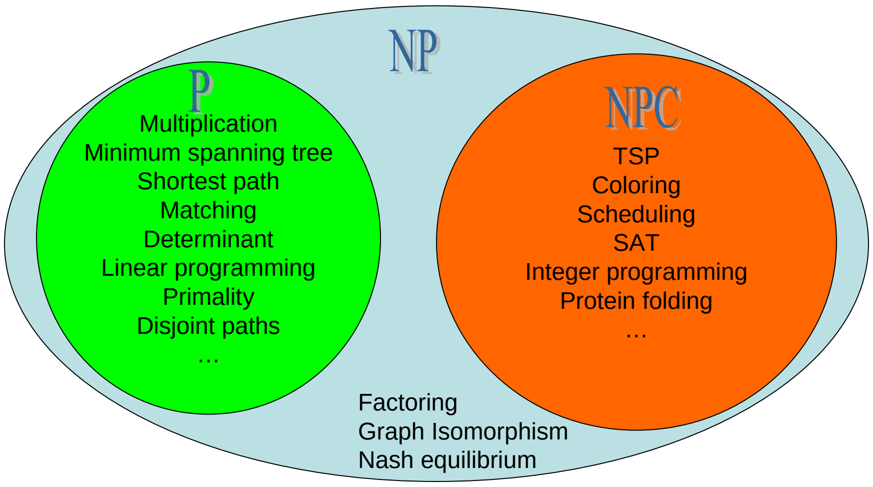
Probability of detecting inconsistencies can be amplified to any desirable level by repeated sampling.

Hardness of Approximation

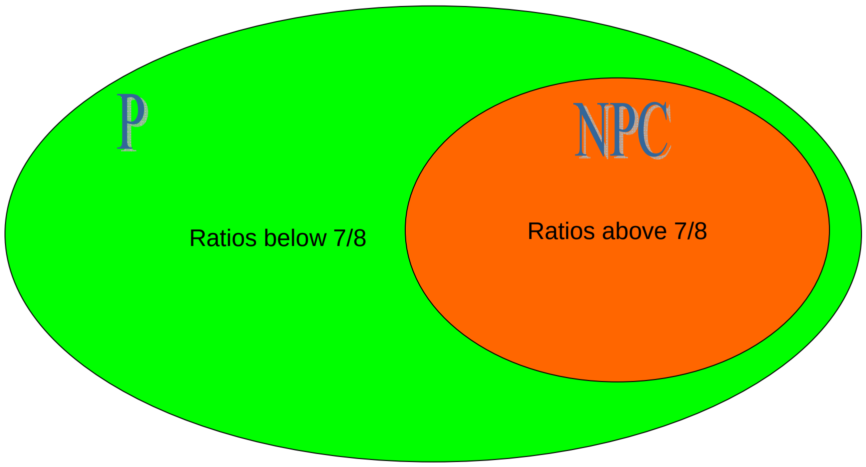
The theory of NP-completeness can help explore the limitations of approximation algorithms.

The PCP theorem plays a key role in reductions that show that improving over certain approximation ratios is NP-hard.

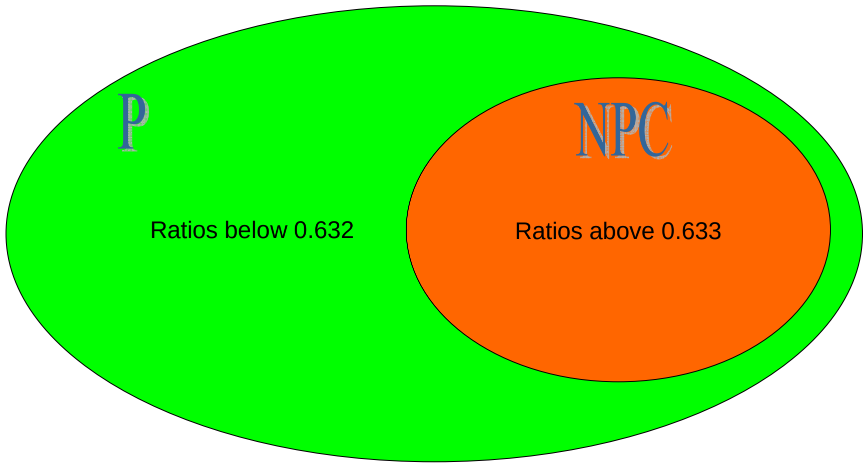
Recall the Classification of Problems



Approximating 3SAT



Approximating Max Coverage



Approximations: State of the Art

For many optimization problems, there is no "middle ground". An approximation ratio is either achievable in polynomial time, or NP-hard. NP-Completeness has amazing explanatory powers for these problems.

Is this true for essentially all optimization problems?

Heuristics

The theory of average case complexity is not as well developed as that of worst case complexity.

Goal: on "most" inputs the algorithm works.

Difficulty: define "most" inputs.

Smoothed complexity: for every input, if the data is perturbed a little at random, the algorithm works almost surely.

Explains the empirical efficiency of the exponential time Simplex algorithm for solving linear programs: it has polynomial smoothed complexity.

Summary

NP-completeness is a beautiful theory, surprisingly powerful, relevant to many areas of science and engineering.

It attracted, still attracts, and will continue to attract a lot of research efforts.

Researchers in Microsoft (theory group and others) are contributing to this theory.

Slides of this talk available on http://research.microsoft.com/theory/feige/

Computational Complexity Disclaimers

Polynomial time algorithms are not always efficient in practice (e.g., high degree polynomials, large hidden constants).

Exponential time algorithms may be efficient for small problems encountered in practice.

Worst case notions. Some NP-hard problems may be easy “on average”.

Asymptotic notions. Not always applicable as is (e.g., playing chess, Rubic’s cube).