

CSCI 355: ALGORITHM DESIGN AND ANALYSIS

10. INTRACTABILITY

- ▶ *poly-time reductions*
- ▶ *P and NP*
- ▶ *NP-completeness*
- ▶ *P vs. NP?*

Algorithm design patterns and antipatterns

Algorithm design patterns.

- Greedy.
- Divide and conquer.
- Dynamic programming.
- Duality.
- Reductions.
- Local search.
- Randomization.

Algorithm design antipatterns.

- NP-completeness. $O(n^k)$ algorithm unlikely.
- PSPACE-completeness. $O(n^k)$ certification algorithm unlikely.
- Undecidability. No algorithm possible.

3

Historical significance: Edmonds' *Paths, Trees, and Flowers*, 1965

2. Digression. An explanation is due on the use of the words "efficient algorithm." First, what I present is a conceptual description of an algorithm and not a particular formalized algorithm or "code."

For practical purposes computational details are vital. However, my purpose is only to show as attractively as I can that there is an efficient algorithm. According to the dictionary, "efficient" means "adequate in operation or performance." This is roughly the meaning I want—in the sense that it is conceivable for maximum matching to have no efficient algorithm. Perhaps a better word is "good."

I am claiming, as a mathematical result, the existence of a *good* algorithm for finding a maximum cardinality matching in a graph.

There is an obvious finite algorithm, but that algorithm increases in difficulty exponentially with the size of the graph. It is by no means obvious whether or not there exists an algorithm whose difficulty increases only algebraically with the size of the graph.



Edmonds

4

Classifying problems according to computational requirements

Q. Which problems will we be able to solve in practice?

A **working definition**. Those with poly-time algorithms.

Turing machine, word RAM, uniform circuits, ...

Theory. Definition is broad and robust.

constants tend to be small, e.g., $3n^2$

Practice. Poly-time algorithms scale to huge problems.

5

Classifying problems according to computational requirements

Q. Which problems will we be able to solve in practice?

A **working definition**. Those with poly-time algorithms.

yes	(probably) no
shortest path	longest path
min cut	max cut
2-satisfiability	3-satisfiability
planar 4-colourability	planar 3-colourability
bipartite vertex cover	vertex cover
matching	3d-matching
primality testing	factoring
linear programming	integer linear programming

6

Classifying problems

Desiderata. Classify problems according to those that can be solved in polynomial time and those that cannot.

Problems that provably require exponential time.

- Given a constant-size program, does it halt in at most k steps?
- Given a board position in an n -by- n generalization of checkers, can black guarantee a win?

input size = $c + \log k$

using forced capture rule



Frustrating news. Huge number of fundamental problems have defied classification for decades.

7

Examples of problems

Satisfiability.

- **SAT.** Given a CNF formula Φ , does it have a satisfying truth assignment?
- **3-SAT.** An instance of SAT where each clause contains exactly 3 literals (and each literal corresponds to a different variable).

Packing and covering.

- **INDEPENDENT-SET.** Given a graph $G = (V, E)$ and an integer k , is there a subset of k (or more) vertices such that no two are adjacent?
- **VERTEX-COVER.** Given a graph $G = (V, E)$ and an integer k , is there a subset of k (or fewer) vertices such that each edge is incident to at least one vertex in the subset?
- **SET-COVER.** Given a set U of elements, a collection S of subsets of U , and an integer k , are there $\leq k$ of these subsets whose union is equal to U ?

12

Examples of problems

Sequencing.

- **HAMILTON-CYCLE.** Given an undirected graph $G = (V, E)$, does there exist a cycle Γ that visits every vertex exactly once?
- **DIRECTED-HAMILTON-CYCLE.** Given a directed graph $G = (V, E)$, does there exist a directed cycle Γ that visits every vertex exactly once?

Colouring.

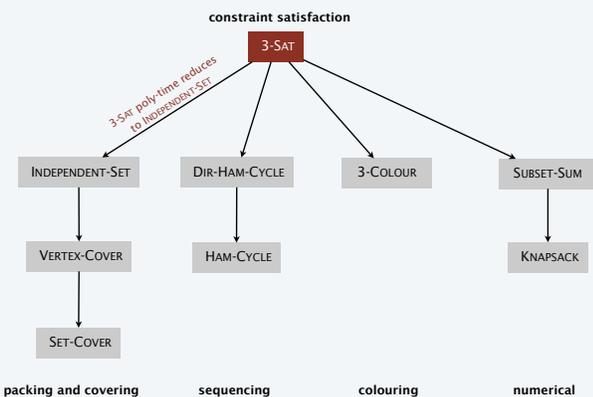
- **3-COLOUR.** Given an undirected graph G , can the vertices be coloured black, white, and blue so that no adjacent vertices have the same colour?

Numerical.

- **SUBSET-SUM.** Given n natural numbers w_1, \dots, w_n and an integer W , is there a subset that adds up to exactly W ?
- **KNAPSACK.** Given $2n$ natural numbers $w_1, \dots, w_n, v_1, \dots, v_n$ and an integer W , is there a subset that maximizes v_i while adding up all values w_i to exactly W ?

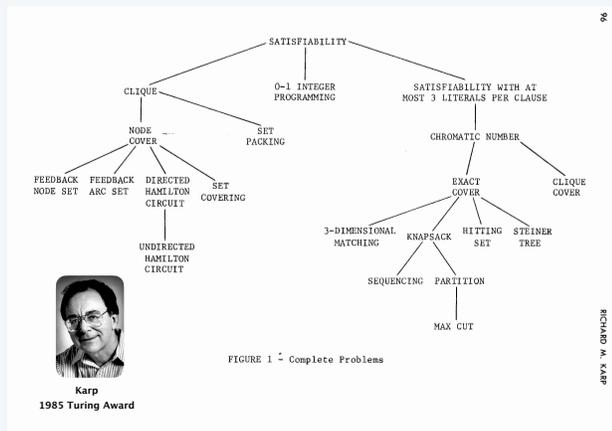
13

Tree of poly-time reductions between problems



14

Karp's Reducibility Among Combinatorial Problems, 1972



CSCI 355: ALGORITHM DESIGN AND ANALYSIS 10. INTRACTABILITY

- ▶ *poly-time reductions*
- ▶ *P and NP*
- ▶ *NP-completeness*
- ▶ *P vs. NP?*

The class P

Decision problems.

- A problem X is a set of strings.
- An instance s of a problem is one string.
- An algorithm A solves problem X : $A(s) = \begin{cases} \text{yes} & \text{if } s \in X \\ \text{no} & \text{if } s \notin X \end{cases}$

Def. Algorithm A runs in **polynomial time** if, for every string s , $A(s)$ terminates in $\leq p(|s|)$ "steps," where $p(\cdot)$ is some polynomial function.

↑
length of s

Def. \mathbf{P} = set of decision problems for which there exists a poly-time algorithm.

↑
on a deterministic
Turing machine

problem PRIMES: { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ... }
instance s : 592335744548702854681
algorithm: Agrawal-Kayal-Saxena (2002)

Some problems in P

P. Set of decision problems for which there exists a poly-time algorithm.

problem	description	poly-time algorithm	yes	no
MULTIPLE	Is x a multiple of y ?	grade-school division	51, 17	51, 16
REL-PRIME	Are x and y relatively prime?	Euclid's algorithm	34, 39	34, 51
PRIMES	Is x prime?	Agrawal-Kayal-Saxena	53	51
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Needleman-Wunsch	niether neither	acgggt ttttta
L-SOLVE	Is there a vector x that satisfies $Ax = b$?	Gauss-Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
U-CONN	Is an undirected graph G connected?	depth-first search		

18

The class NP

Def. An algorithm $C(s, t)$ is a **certifier** for problem X if for every string $s : s \in X$ iff there exists a string t such that $C(s, t) = \text{yes}$.

↑
"certificate" or "witness"

Def. **NP** = set of decision problems for which there exists a poly-time certifier.

- $C(s, t)$ is a poly-time algorithm.
- Certificate t is of polynomial size: $|t| \leq p(|s|)$ for some polynomial $p(\cdot)$.

problem COMPOSITES:	{ 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, ... }
instance s :	437669
certificate t :	541 ← 437,669 = 541 × 809
certifier $C(s, t)$:	grade-school division

19

Certifiers and certificates: satisfiability

SAT. Given a CNF formula Φ , does it have a satisfying truth assignment?

3-SAT. SAT where each clause contains exactly 3 literals.

Certificate. An assignment of truth values to the Boolean variables.

Certifier. Checks that each clause in Φ has at least one true literal.

instance s	$\Phi = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4)$
certificate t	$x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}, x_4 = \text{false}$

Conclusions. SAT \in NP, 3-SAT \in NP.

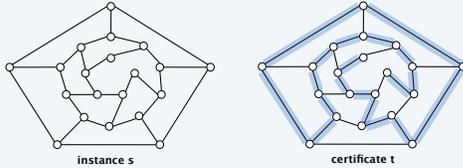
20

Certifiers and certificates: Hamiltonian path

HAMILTON-PATH. Given an undirected graph $G = (V, E)$, does there exist a simple path P that visits every vertex?

Certificate. A permutation π of the n vertices.

Certifier. Checks that π contains each vertex in V exactly once, and that G contains an edge between each pair of adjacent vertices.



Conclusion. HAMILTON-PATH \in NP.

21

Some problems in NP

NP. Set of decision problems for which there exists a poly-time certifier.

problem	description	poly-time algorithm	yes	no
L-SOLVE	Is there a vector x that satisfies $Ax = b$?	Gauss-Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
COMPOSITES	Is x composite?	Agrawal-Kayal-Saxena	51	53
FACTOR	Does x have a nontrivial factor less than y ?	???	(56159, 50)	(55687, 50)
SAT	Given a CNF formula, does it have a satisfying truth assignment?	???	$\neg x_1 \vee \neg x_2 \vee \neg x_3$ $x_1 \vee \neg x_2 \vee x_3$ $\neg x_1 \vee \neg x_2 \vee x_3$	$\neg x_2$ x_2 $\neg x_1 \vee x_2$
HAMILTON-PATH	Is there a simple path between u and v that visits every vertex?	???		

22

Significance of NP

NP. Set of decision problems for which there exists a poly-time certifier.

“ NP captures vast domains of computational, scientific, and mathematical endeavours, and seems to roughly delimit what mathematicians and scientists have been aspiring to compute feasibly. ”

— Christos Papadimitriou

25

The classes P, NP, and EXP

P. Set of decision problems for which there exists a poly-time algorithm.

NP. Set of decision problems for which there exists a poly-time certifier.

EXP. Set of decision problems for which there exists an exp-time algorithm.

Proposition. $P \subseteq NP$.

Pf. Consider any problem $X \in P$.

- By definition, there exists a poly-time algorithm $A(s)$ that solves X .
- Certificate is $t = \epsilon$, certifier is $C(s, t) = A(s)$. ▀

Proposition. $NP \subseteq EXP$.

Pf. Consider any problem $X \in NP$.

- By definition, there exists a poly-time certifier $C(s, t)$ for X where a certificate t satisfies $|t| \leq p(|s|)$ for some polynomial $p(\cdot)$.
- To solve the instance s , run $C(s, t)$ on all strings t with $|t| \leq p(|s|)$.
- Return *yes* iff $C(s, t)$ returns *yes* for any of these potential certificates. ▀

Fact. $P \neq EXP \Rightarrow$ either $P \neq NP$, or $NP \neq EXP$, or both.

26

CSCI 355: ALGORITHM DESIGN AND ANALYSIS 10. INTRACTABILITY

▶ *poly-time reductions*

▶ *P and NP*

▶ *NP-completeness*

▶ *P vs. NP?*

NP-completeness

NP-completeness. A problem $Y \in NP$ is NP-complete if it has the property that for every problem $X \in NP$, $X \leq_p Y$.

Proposition. Suppose $Y \in NP$ -complete. Then $Y \in P$ iff $P = NP$.

Pf.

[\Leftarrow] If $P = NP$, then $Y \in P$ because $Y \in NP$.

[\Rightarrow] Suppose $Y \in P$.

- Consider any problem $X \in NP$. Since $X \leq_p Y$, we have $X \in P$.
- This implies $NP \subseteq P$.
- We already know $P \subseteq NP$. Thus $P = NP$. ▀

Fundamental question. Are there any "natural" NP-complete problems?

28

Some NP-complete problems

Basic classes of NP-complete problems and examples.

- Packing/covering problems: SET-COVER, VERTEX-COVER, INDEPENDENT-SET.
- Constraint satisfaction problems: SAT, 3-SAT, CIRCUIT-SAT.
- Sequencing problems: HAMILTON-CYCLE, TSP.
- Partitioning problems: 3-COLOUR, 3D-MATCHING.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Practice. Most NP problems are known to be either in P or NP-complete.

"NP-intermediate" problems? FACTOR, DISCRETE-LOG, GRAPH-ISOMORPHISM, ...

Theorem. [Ladner 1975] Unless $P = NP$, there exist problems in NP that are neither in P nor NP-complete.

On the Structure of Polynomial Time Reducibility

RICHARD E. LADNER
University of Washington, Seattle, Washington

34

More hard computational problems

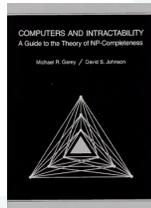
M. R. Garey and D. S. Johnson. *Computers and Intractability.*

- Appendix includes over 300 NP-complete problems.
- Most cited reference in computer science literature.

Most Cited Computer Science Citations

This list is generated from documents in the Citeseer database as of January 17, 2013. This list is automatically generated and may contain errors. The list is generated in batch mode and citation counts may differ from those currently in the Citeseer database, since the database is continuously updated.

1. M R Garey, D S Johnson
Computers and intractability: A Guide to the Theory of NP-Completeness 1979
6965
2. T Cormen, C E Leiserson, R Rivest
Introduction to Algorithms 1990
7210
3. V N Vapnik
The Nature of Statistical Learning Theory 1998
6580
4. A P Dempster, N M Laird, D B Rubin
Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 1977
6982
5. I. Cover, J Thomas
Elements of Information Theory 1991
6075
6. D E Goldberg
Genetic Algorithms in Search, Optimization, and Machine Learning, 1989
6988
7. J Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference 1988
5082
8. C C Galletta, R Heem, R Johnson, J Masnick
Design Patterns: Elements of Reusable Object-Oriented Software 1995
4214
9. C E Shannon
A mathematical theory of communication Bell Syst. Tech. J. 1948
4118
10. J R Quinlan
S.I.G. Programs for Machine Learning 1993
4018



35

More hard computational problems

Aerospace engineering. Optimal mesh partitioning for finite elements.

Biology. Phylogeny reconstruction.

Chemical engineering. Heat exchanger network synthesis.

Chemistry. Protein folding.

Civil engineering. Equilibrium of urban traffic flow.

Economics. Computation of arbitrage in financial markets with friction.

Electrical engineering. VLSI layout.

Environmental engineering. Optimal placement of contaminant sensors.

Financial engineering. Minimum risk portfolio of given return.

Game theory. Nash equilibrium that maximizes social welfare.

Mathematics. Given integer a_1, \dots, a_n , compute $\int_0^{2\pi} \cos(a_1\theta) \times \cos(a_2\theta) \times \dots \times \cos(a_n\theta) d\theta$

Mechanical engineering. Structure of turbulence in sheared flows.

Medicine. Reconstructing 3d shape from biplane angiogram.

Operations research. Traveling salesperson problem.

Physics. Partition function of 3d Ising model.

Politics. Shapley-Shubik voting power.

Recreation. Versions of Sudoku, Checkers, Minesweeper, Tetris, Rubik's Cube.

Statistics. Optimal experimental design.

36

Other possible outcomes

$P = NP$, but with a $\Omega(n^{100})$ algorithm for 3-SAT.

$P \neq NP$, but with a $O(n^{\log^* n})$ algorithm for 3-SAT.

$P = NP$ is independent of ZFC axiomatic set theory.

" It will be solved by either 2048 or 4096. I am currently somewhat pessimistic. The outcome will be the truly worst case scenario: namely that someone will prove $P = NP$ because there are only finitely many obstructions to the opposite hypothesis; hence there exists a polynomial time solution to SAT but we will never know its complexity! "

— Donald Knuth (2002)



44

Other possible outcomes

" I feel that theoretical computer scientists should devote a constant fraction of their lives to trying to resolve the P vs. NP question.

I personally spend a few days each year thinking about it. I've proven (at least twice) that NP does not equal co-NP (and hence P does not equal NP). I've also proven (also at least twice) that NP equals co-NP.

My most recent proof that NP does not equal co-NP occurred about a week ago as I write this, and the proof survived for about half an hour (not quite long enough for me to run it by someone else). My longest-surviving proof that NP does not equal co-NP survived for about 3 days and fooled some very smart people into believing it. "

— Ronald Fagin (2002)



45

Millennium prize

Millennium Problems. \$1 million for a resolution to the P vs. NP problem.

The only Millennium Problem relating to CS!

- Birch and Swinnerton-Dyer conjecture
- Hodge conjecture
- Navier-Stokes existence and smoothness
- **P vs. NP problem**
- Poincaré conjecture (solved)
- Riemann hypothesis
- Yang-Mills existence and mass gap



Clay Mathematics Institute
Dedicated to increasing and disseminating mathematical knowledge

Millennium Problems

In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven Prize Problems. The Scientific Advisory Board of CMI has selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a \$7 million prize fund for the solution to these problems, with \$1 million allocated to each. During the Millennium Meeting held on May 24, 2000 at the Collège de France, Timothy Gowers presented a lecture entitled *The Importance of Mathematics*, aimed for the general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- P vs. NP
- Poincaré Conjecture
- Riemann Hypothesis
- Yang-Mills Theory
- Mass
- Millennium Media Videos

46

