The Design and Analysis of Algorithms

Lecture 18 Intractability III

Zhenbo Wang

Department of Mathematical Sciences, Tsinghua University

Content

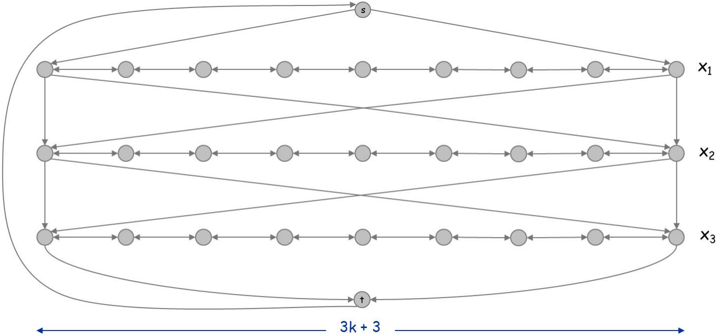
3-SAT Reduces to Directed Hamiltonian Cycle

Theorem 1

3-SAT $\leq_P$ *DIR-HAM-CYCLE.*

Pf. Given an instance Φ of *3-SAT* with n variables x_i and k clauses, we construct an instance of *DIR-HAM-CYCLE* that has a Hamilton cycle iff Φ is satisfiable.

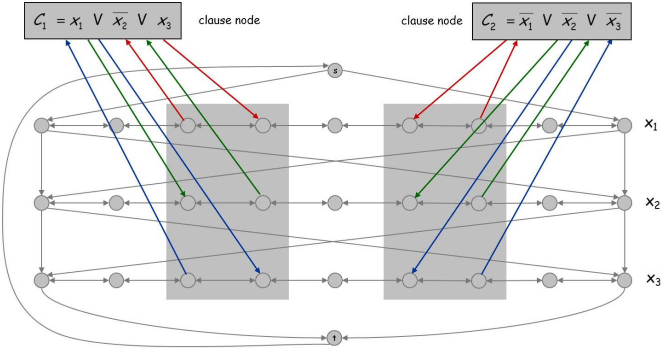
Construction. Create graph that has 2^n Hamilton cycles correspond to 2^n possible truth assignments.



3-SAT Reduces to Directed Hamiltonian Cycle

Intuition: traverse path i from left to right $\Leftrightarrow$ set variable $x_i =$ true.

For each clause, add a node and 6 edges.



3-SAT Reduces to Directed Hamiltonian Cycle

Lemma 2

Φ is satisfiable iff G has a Hamilton cycle.

Pf. $\Rightarrow$ Suppose *3-SAT* instance has satisfying assignment x^*.

Define Hamilton cycle in G as follows:

if $x_i^* = true$, traverse row i from left to right;

if $x_i^* = false$, traverse row i from right to left.

For each clause C_j , there will be at least one row i in which we are going in "correct" direction to join clause node C_j into cycle (exactly once).

3-SAT Reduces to Directed Hamiltonian Cycle

$\Leftarrow$ Suppose G has a Hamilton cycle Γ.

If Γ enters clause node C_j, it must depart on mate edge.

Set $x_i^* = \textit{true}$ iff Γ traverses row i left to right.

Since Γ visits each clause node C_j, at least one of the paths is traversed in "correct" direction, and each clause is satisfied. $\square$

3-SAT Reduces to Longest Path

- *LONGEST-PATH*. Given a directed graph $G = (V, E)$, does there exists a simple path consisting of at least k edges?

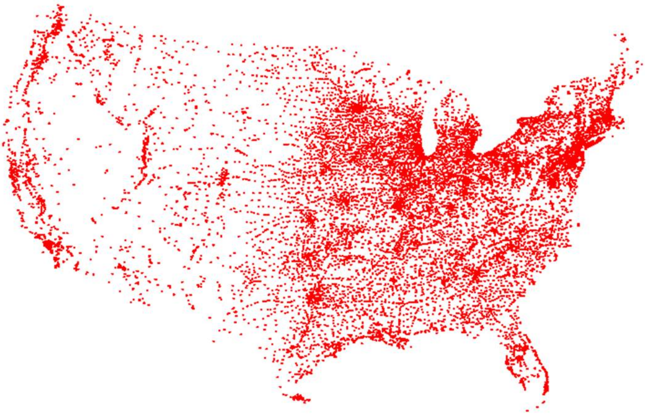
Theorem 3

3-SAT $\leq_P$ LONGEST-PATH.

Pf. Redo proof for DIR-HAM-CYCLE, ignoring back-edge from t to s. $\square$

Traveling Salesperson Problem

- *TSP*. Given a set of n cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$?



All 13,509 cities in US with a population of at least 500
Reference: http://www.tsp.gatech.edu

Hamilton Cycle Reduces to *TSP*

Theorem 4

HAM-CYCLE $\leq_P$ *TSP*.

Pf. Given instance $G = (V, E)$ of HAM-CYCLE, create n cities with distance function

$$d(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{if } (u, v) \notin E \end{cases}$$

- *TSP* instance has tour of length $\leq n$ iff G has a Hamilton cycle. $\square$

Remark. *TSP* instance satisfies triangle inequality: $d(u, w) \leq d(u, v) + d(v, w)$.

3-Dimensional Matching

- *3D-MATCHING*. Given 3 disjoint sets X, Y, and Z, each of size n and a set $T \subseteq X \times Y \times Z$ of triples, does there exist a set of n triples in T such that each element of $X \cup Y \cup Z$ is in exactly one of these triples?

Theorem 5

3-SAT $\leq_P$ 3D-MATCHING.

- Refer to the textbook.

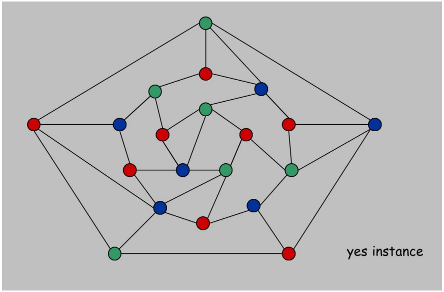
3-Colorability

- *3-COLOR*. Given an undirected graph G, can the nodes be colored and blue so that no adjacent nodes have the same color?

Theorem 6

3-SAT $\leq_P$ 3-COLOR.

- Refer to the textbook.



Subset Sum

- *SUBSET-SUM*. Given natural numbers $w_1, \cdots, w_n$ and an integer W, is there a subset that adds up to exactly W?

Ex. $\{1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344\}$, $W = 3754$.

Yes. $1 + 16 + 64 + 256 + 1040 + 1093 + 1284 = 3754$.

Remark. With arithmetic problems, input integers are encoded in binary. Poly-time reduction must be polynomial in binary encoding.

Theorem 7

3-SAT $\leq_P$ *SUBSET-SUM.*

Pf. Given an instance Φ of *3-SAT*, we construct an instance of *SUBSET-SUM* that has solution iff Φ is satisfiable.

3-Satisfiability Reduces to Subset Sum

- *Construction*. Given *3-SAT* instance Φ with n variables and k clauses, form $2n + 2k$ decimal integers, each of $n + k$ digits:

$$C_1 = \bar{x} \vee y \vee z$$
$$C_2 = x \vee \bar{y} \vee z$$
$$C_3 = \bar{x} \vee \bar{y} \vee \bar{z}$$

	x	y	z	C_1	C_2	C_3	
x	1	0	0	0	1	0	100,010
¬ x	1	0	0	1	0	1	100,101
y	0	1	0	1	0	0	10,100
¬ y	0	1	0	0	1	1	10,011
z	0	0	1	1	1	0	1,110
¬ z	0	0	1	0	0	1	1,001
	0	0	0	1	0	0	100
	0	0	0	2	0	0	200
	0	0	0	0	1	0	10
	0	0	0	0	2	0	20
	0	0	0	0	0	1	1
	0	0	0	0	0	2	2
W	1	1	1	4	4	4	111,444

dummies to get clause columns to sum to 4

3-Satisfiability Reduces to Subset Sum

Lemma 8

Φ is satisfiable iff there exists a subset that sums to W.

Pf. $\Rightarrow$ Suppose Φ is satisfiable.

Choose integers corresponding to each *true* literal.

Since Φ is satisfiable, each C_j digit sums to at least 1 from x_i rows.

Choose dummy integers to make clause digits sum to 4.

$\Leftarrow$ Suppose there is a subset that sums to W.

Digit x_i forces subset to select either row x_i or $\neg x_i$ (but not both).

Digit C_j forces subset to select at least one literal in clause.

Assign $x_i = true$ iff row x_i selected. $\square$

Partition Problem

- *PARTITION*. Given natural numbers $v_1, \cdots, v_m$, can they be partitioned into two subsets that add up to the same value $1/2 \sum_i v_i$?

Theorem 9

SUBSET-SUM $\leq_P$ *PARTITION*.

Pf. Let $W, w_1, \cdots, w_n$ be an instance of *SUBSET-SUM*.

Create instance of *PARTITION* with $m = n + 2$ elements:

$v_1 = w_1, v_2 = w_2, \cdots, v_n = w_n, v_{n+1} = 2\sum_i w_i - W, v_{n+2} = \sum_i w_i + W$.

There exists a subset that sums to W iff there exists a partition since elements v_{n+1} and v_{n+2} cannot be in the same partition. $\square$

Scheduling with Release Times

- *SCHEDULE*. Given a set of n jobs with processing time t_j, release time r_j, and deadline d_j, is it possible to schedule all jobs on a single machine such that job j is processed with a contiguous slot of t_j time units in the interval $[r_j, d_j]$?

Theorem 10

SUBSET-SUM $\leq_P$ Scheduling.

Pf. Let $W, w_1, \cdots, w_n$ be an instance of *SUBSET-SUM*.

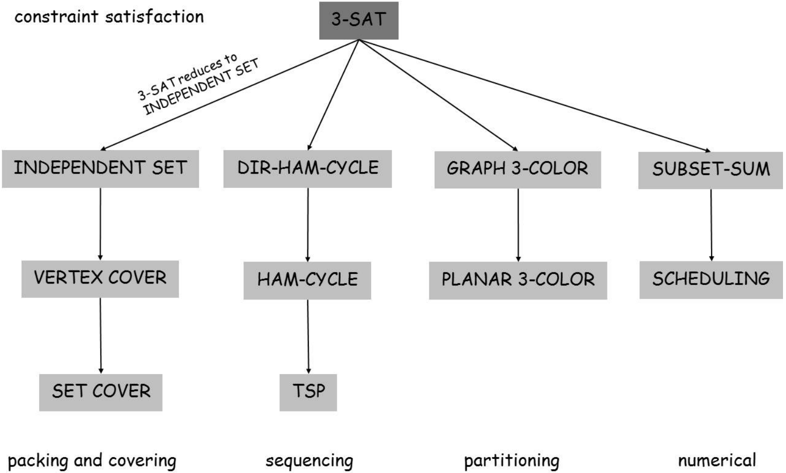
Create instance of *Scheduling* that is feasible iff there exists a subset that sums to exactly W:

Create n jobs with processing time $t_j = w_j$, release time $r_j = 0$ and deadline $\sum_{i=1}^{n} w_i + 1$;

Create job 0 with $t_0 = 1$, release time $r_0 = W$, and deadline $d_0 = W + 1$.

Subset that sums to W iff there exists a feasible schedule. $\square$

Polynomial-Time Reductions



- *Observation*. All these problems are *NP-complete* and polynomial reduce to one another!

Karp's 21 NP-Complete Problems

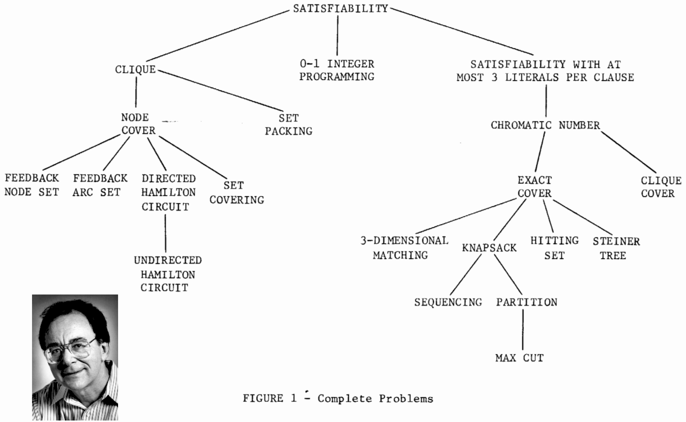


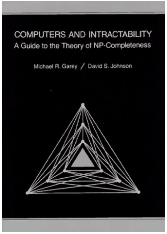
FIGURE 1 - Complete Problems

RICHARD M. KARP

Dick Karp (1972)
1985 Turing Award

NP-complete Problems

M. R. Gary and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979.



P, NP and Beyond

- *Practice*. Most *NP* problems are known to be either in *P* or *NP-complete*.
- *Notable exceptions*. *FACTOR*, *GRAPH-ISOMORPHISM*, *NASH-EQUILIBRIUM*.

Theorem 11 (Ladner 1975)

Unless $P = NP$, there exist problems in NP that are neither in P nor NP-complete.

- Many complexity classes: *co-NP*, *polynomial hierarchy*, *PSPACE*, *BPP*, *IP* $\cdots$

Homework

- Read the proof on the reductions of *3-dimensional matching* and *3-colorability*.
- Exercises 7 & 27 in Chapter 8.