CMP(N)302: Design and Analysis of Algorithms

Lecture 10: NP Completeness

Ahmed Hamdy

Computer Engineering Department

Cairo University

Fall 2019

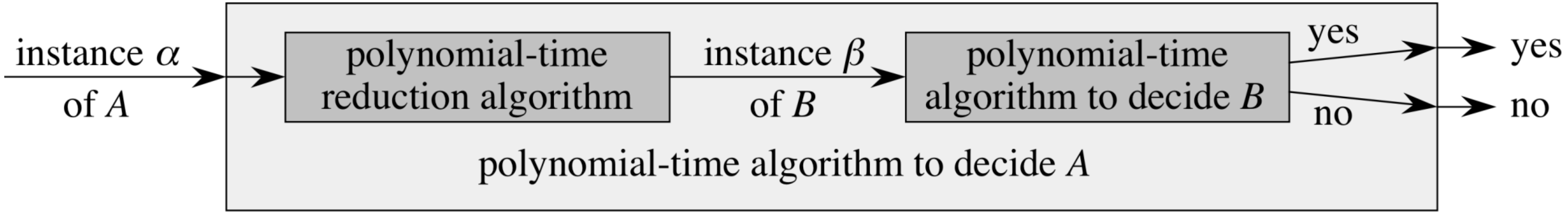
What is NP-complete?

- NP stands for "Non-deterministic Polynomial-time"
- All algorithms studied so far are *polynomial-time algorithms*, a.k.a $O(n^k)$
- Similar problems but P vs NP-complete:
  - – Shortest vs longest simple paths (not DAG)
  - – Euler tour vs Hamiltonian cycle
    - Euler tour: traverses each *edge* once, $O(E)$
    - Hamiltonian cycle: traverses each *vertex* once, NP-complete
  - – 2-CNF satisfiability $((\overline{x_1} \vee x_2) \wedge (x_1 \vee x_3)\ldots)$ vs. 3-CNF satisfiability $((\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_3 \vee x_4)\ldots)$

Decision vs. optimization problems

- Optimization problem: solution achieves min/max
- Decision problem: solution is “yes” or “no”
- Decision problem related to (single-pair) shortest-path:
  - Does a path exist from u to v consisting of at most k edges?
- If an optimization problem is easy, its related decision problem is easy as well.
- If evidence exists that a decision problem is hard, means optimization problem is hard.

Reduction



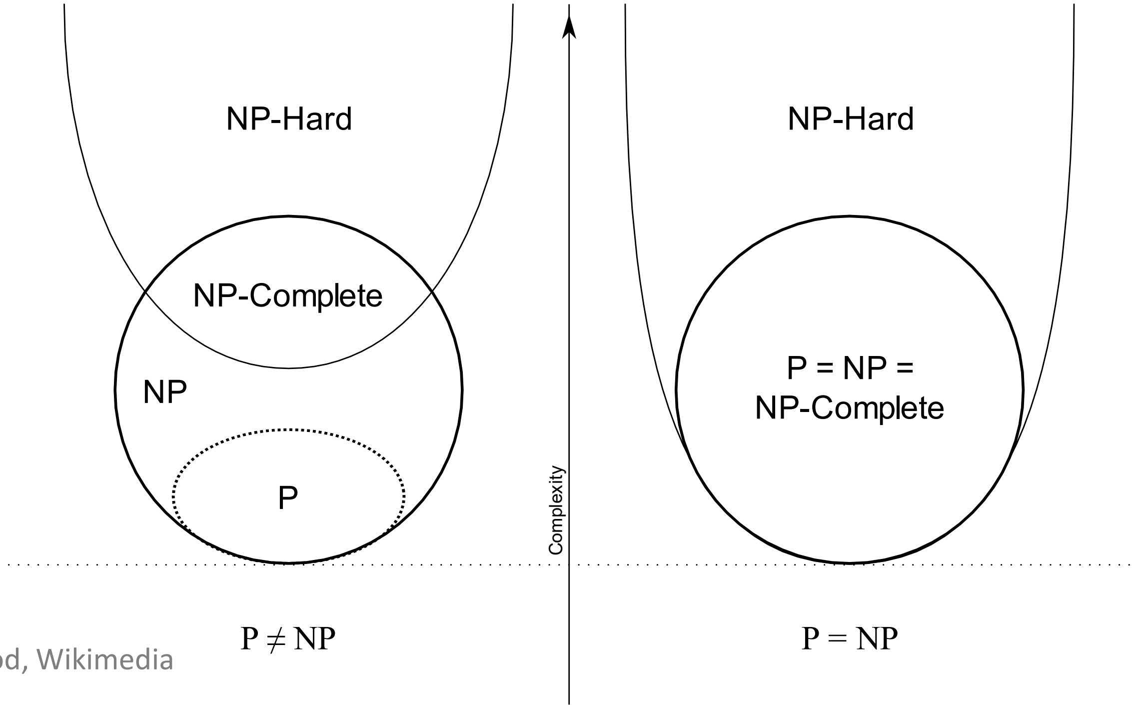
- Requirements:
  - Transformation takes polynomial time.
  - Answers are the same. Answer for A is “yes” if and only if the answer for B is also “yes.”
- If we want to prove that a problem is NP-compete, and another NP-complete is a proven one. Which to reduce to the other?

Polynomial-time verification

- Given a solution, even for an NP-complete problem. It can be verified in polynomial-time.
- Problems with such property are in *complexity class NP.*
- Examples:
  - Verifying a solution for 3-CNF satisfiability
  - Verifying a solution for Hamiltonian cycle
- Problems in *P* are of course verifiable in polynomial-time.

Complexity classes

- P: class of decision problems that can be solved in polynomial time.
- NP: Class of decision problems which their solutions can be *verified* in polynomial time. $P \subseteq NP$.
- NP-hard: Class of decision problems which are at least as hard as the hardest problems in NP.
- NP-complete: Class of decision problems which contains the hardest problems in NP.



Courtesy of Behnam Esfahbod, Wikimedia

Circuit Satisfiability

- First problem proven to be NP-complete
- Will be used to prove all other problems to be NP-complete by reducibility

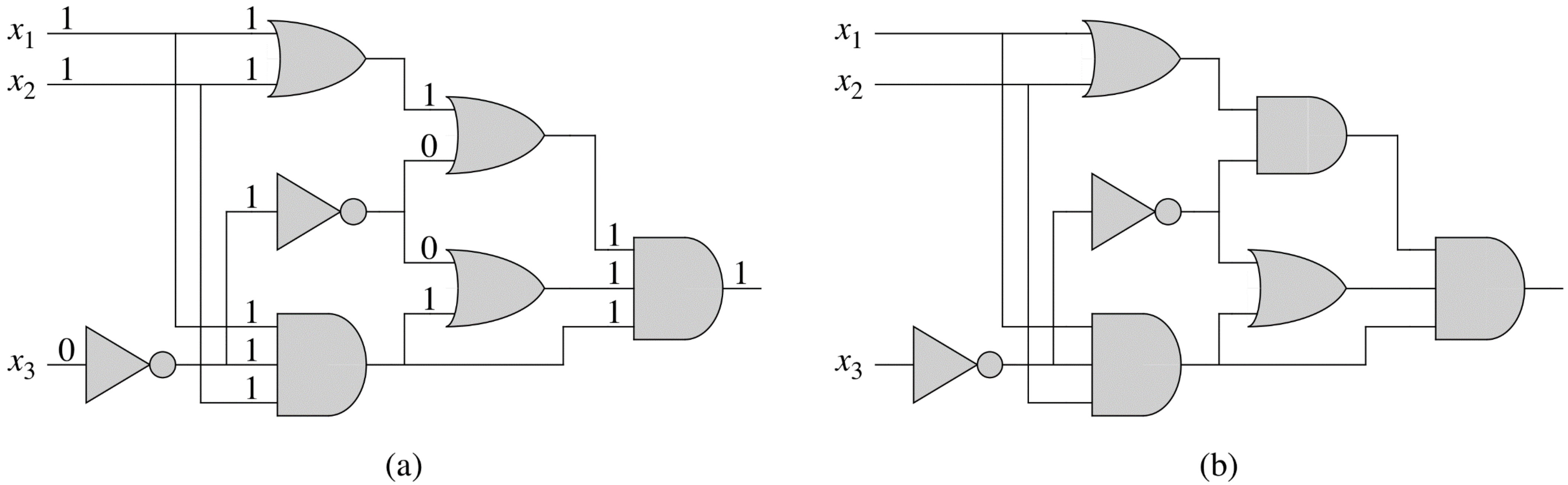
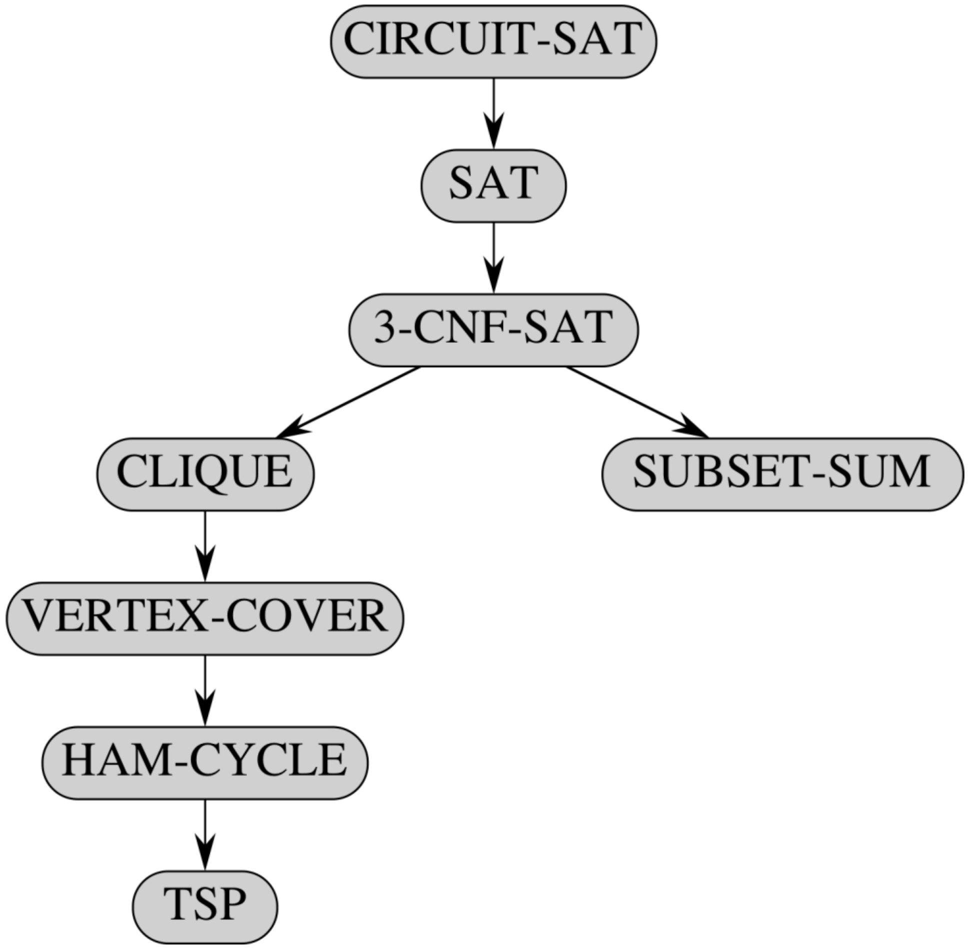


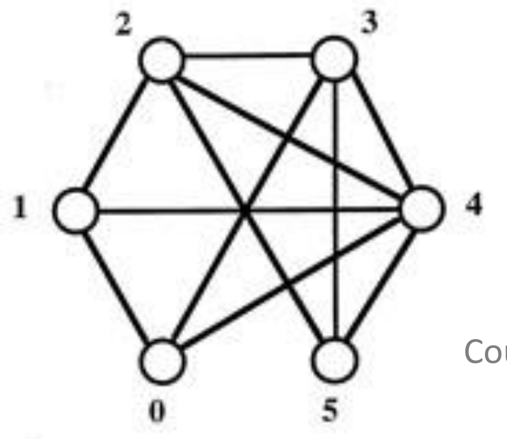
Figure 34.8 Two instances of the circuit-satisfiability problem. **(a)** The assignment $\langle x_1 = 1, x_2 = 1, x_3 = 0\rangle$ to the inputs of this circuit causes the output of the circuit to be 1. The circuit is therefore satisfiable. **(b)** No assignment to the inputs of this circuit can cause the output of the circuit to be 1. The circuit is therefore unsatisfiable.

Reducible problems



Clique problem

- Find a subset of vertices of a graph with maximum size such that all pairs of these vertices are connected by an edge.
- As a decision problem, ask whether a clique of size k exists in the graph.
- What is the maximal clique size for this graph?



Courtesy of Qi Ouyang

- To prove that it is NP-complete, we will reduce 3-CNF-SAT to CLIQUE.

3-CNF-SAT → CLIQUE

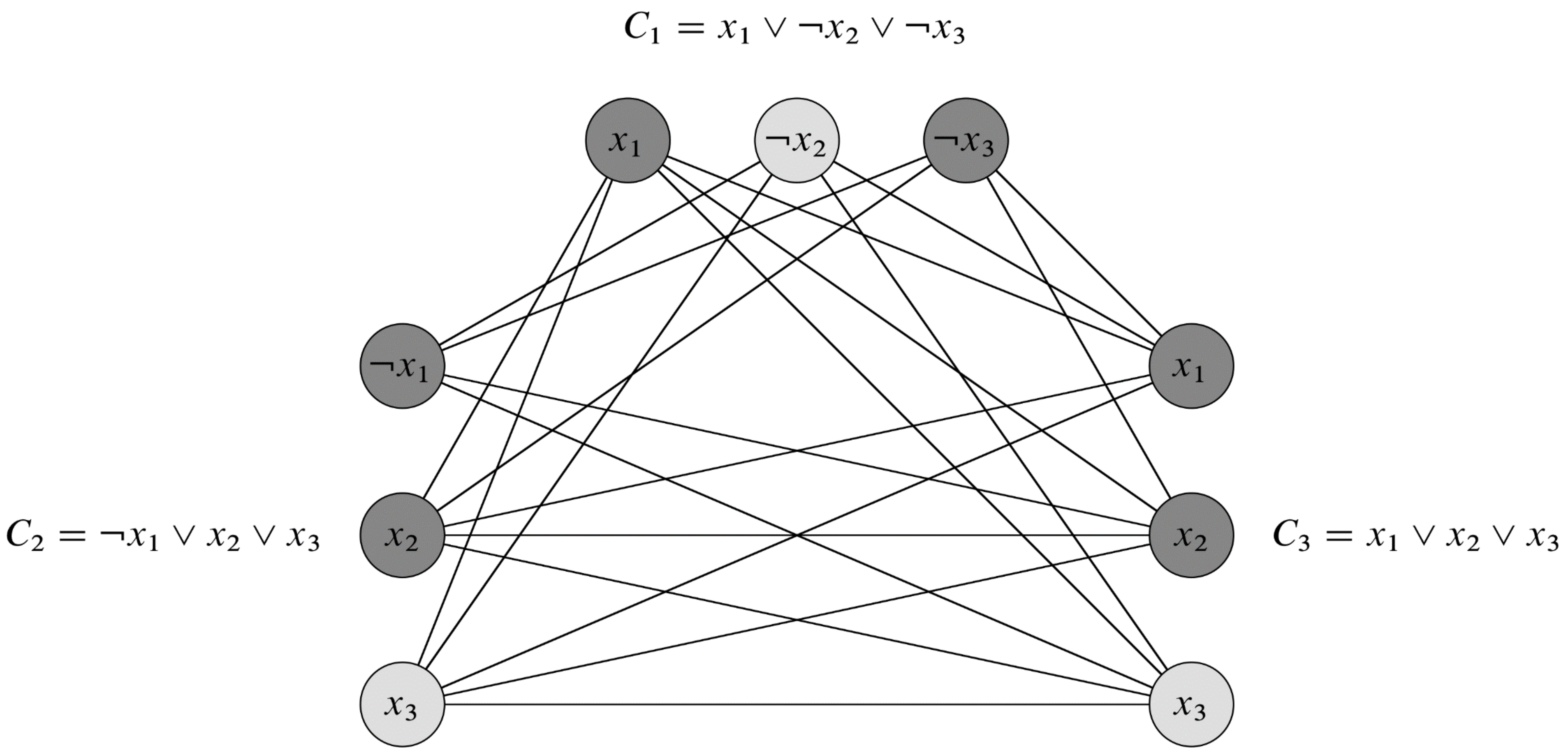
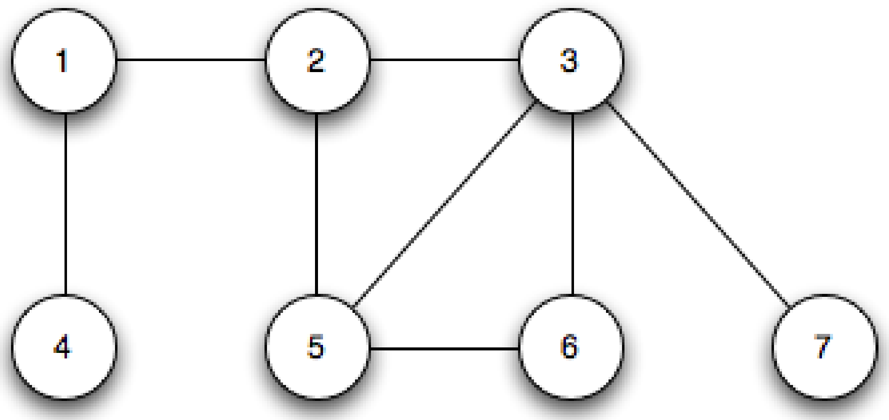


Figure 34.14 The graph G derived from the 3-CNF formula $\phi = C_1 \wedge C_2 \wedge C_3$, where $C_1 = (x_1 \vee \neg x_2 \vee \neg x_3)$, $C_2 = (\neg x_1 \vee x_2 \vee x_3)$, and $C_3 = (x_1 \vee x_2 \vee x_3)$, in reducing 3-CNF-SAT to CLIQUE. A satisfying assignment of the formula has $x_2 = 0$, $x_3 = 1$, and x_1 either 0 or 1. This assignment satisfies C_1 with $\neg x_2$, and it satisfies C_2 and C_3 with x_3, corresponding to the clique with lightly shaded vertices.

Vertex-Cover problem

- Find a subset of vertices of a graph with minimum size such that every vertex “covers” its incident edge.
- As a decision problem, ask whether a vertex-cover with size k exists in the graph.
- What is the optimal vertex-cover for this graph?



Courtesy of David Babcock

CLIQUE → VERTEX-COVER

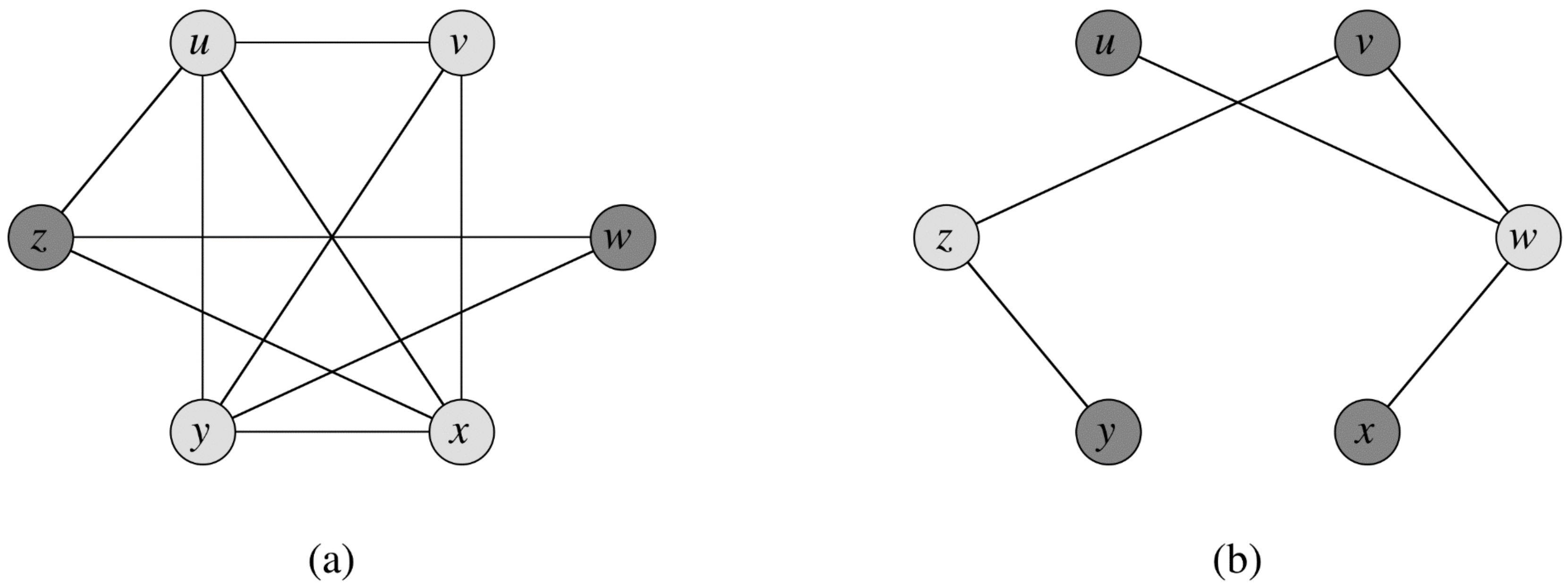


Figure 34.15 Reducing CLIQUE to VERTEX-COVER. **(a)** An undirected graph $G = (V, E)$ with clique $V' = \{u, v, x, y\}$. **(b)** The graph $\overline{G}$ produced by the reduction algorithm that has vertex cover $V - V' = \{w, z\}$.

TSP

- Traveling Salesman Problem
- Find the minimum cost (sum of edges) to visit n-cities, each city visited exactly once and finishing with the first city.

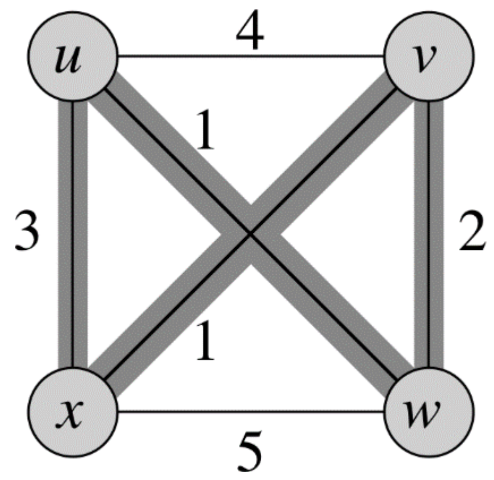


Figure 34.18 An instance of the traveling-salesman problem. Shaded edges represent a minimum-cost tour, with cost 7.

HAM-CYCLE → TSP

To prove that TSP is NP-hard, we show that HAM-CYCLE $\leq_P$ TSP. Let $G = (V, E)$ be an instance of HAM-CYCLE. We construct an instance of TSP as follows. We form the complete graph $G' = (V, E')$, where $E' = \{(i, j) : i, j \in V$ and $i \neq j\}$, and we define the cost function c by

$$c(i, j) = \begin{cases} 0 & \text{if } (i, j) \in E , \\ 1 & \text{if } (i, j) \notin E . \end{cases}$$

Subset-sum problem

- Find a subset of integers picked from a set of positive integers such that their sum equals to a given value t.

if $S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$ and $t = 138457$, then the subset $S' = \{1, 2, 7, 98, 343, 686, 2409, 17206, 117705\}$ is a solution.

- DP algorithm exists to solve in pseudo-polynomial-time for small problem instances.

3-CNF-SAT → SUBSET-SUM

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	=	1	0	0	1	0	0	1
v'_1	=	1	0	0	0	1	1	0
v_2	=	0	1	0	0	0	0	1
v'_2	=	0	1	0	1	1	1	0
v_3	=	0	0	1	0	0	1	1
v'_3	=	0	0	1	1	1	0	0
s_1	=	0	0	0	1	0	0	0
s'_1	=	0	0	0	2	0	0	0
s_2	=	0	0	0	0	1	0	0
s'_2	=	0	0	0	0	2	0	0
s_3	=	0	0	0	0	0	1	0
s'_3	=	0	0	0	0	0	2	0
s_4	=	0	0	0	0	0	0	1
s'_4	=	0	0	0	0	0	0	2
t	=	1	1	1	4	4	4	4

Figure 34.19 The reduction of 3-CNF-SAT to SUBSET-SUM. The formula in 3-CNF is $\phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$, where $C_1 = (x_1 \vee \neg x_2 \vee \neg x_3)$, $C_2 = (\neg x_1 \vee \neg x_2 \vee \neg x_3)$, $C_3 = (\neg x_1 \vee \neg x_2 \vee x_3)$, and $C_4 = (x_1 \vee x_2 \vee x_3)$. A satisfying assignment of ϕ is $\langle x_1 = 0, x_2 = 0, x_3 = 1 \rangle$. The set S produced by the reduction consists of the base-10 numbers shown; reading from top to bottom, $S = \{1001001, 1000110, 100001, 101110, 10011, 11100, 1000, 2000, 100, 200, 10, 20, 1, 2\}$. The target t is 1114444. The subset $S' \subseteq S$ is lightly shaded, and it contains v'_1, v'_2, and v_3, corresponding to the satisfying assignment. It also contains slack variables s_1, s'_1, s'_2, s_3, s_4, and s'_4 to achieve the target value of 4 in the digits labeled by C_1 through C_4.