NP-Completeness

How to show a problem A is NP-complete

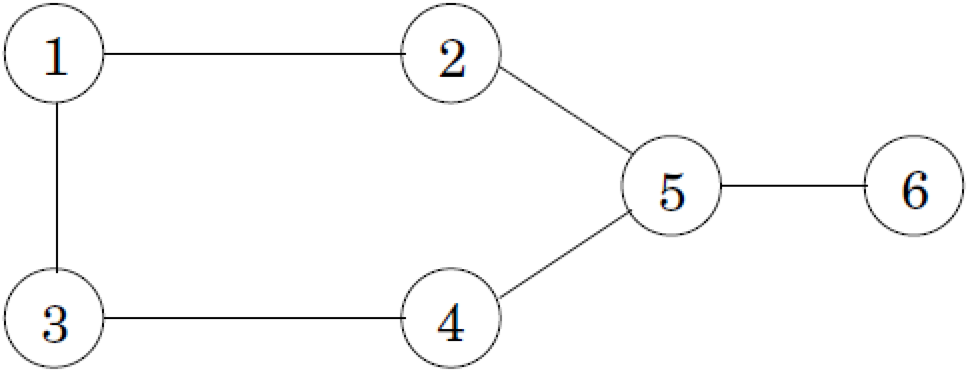
- Show that A is NP
  - Show that if the solution is 1(yes), there exists a polynomial-size certificate that we can verify in polynomial time.
- Show that A is NP-hard
  - Reduce a known NP-hard problem B to A.
    1. Describe a polynomial-time algorithm to transform an input x to B to an input $f(x)$ to A.
    2. Show that the solution of A for $f(x)$ is 1(yes) if and only if the solution of B for x is 1(yes).

Decision problem vs. optimization problem

- ***Decision problem*** : answer is either “yes (1)” or “no (0)”.
- ***Optimization problem*** : want to find a feasible solution with the best value.
- NP-completeness applies directly to decision problem.
- How can we show an optimization problem is hard?
- Cast an optimization problem into a decision problem.
  - Ex) shortest-path problem : ask whether there is a path consisting of at most k edges.
  - Ex) MST : ask whether there is a spanning tree with a cost at most k.
- We can solve decision problem by solving optimization problem. (decision problem is reduced to optimization problem). So, decision problem is at least no harder than optimization problem.
- By showing a decision problem is NPC, we provide an evidence that the optimization problem is hard.

Independent Set

- Let $G = (V, E)$ be an undirected graph.
- A subset $W \subseteq V$ is *independent* if none of the vertices in W are adjacent.
- ***The maximum independent set problem*** asks for the size of the largest independent set in a given graph.
- Given G and an integer k, the independent set problem (IND-SET) asks whether or not there is an independent set of k or more vertices.
- We'll show that IND-SET is NP-hard, using a reduction from 3-SAT.

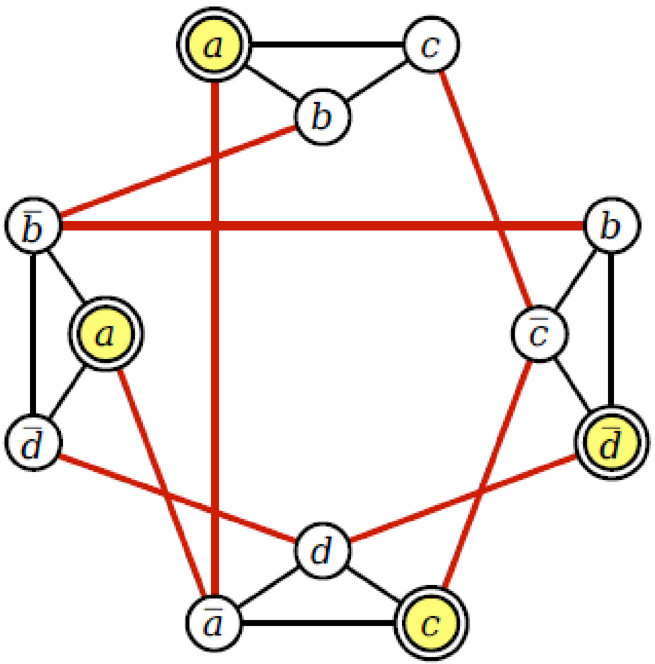


Reduction from 3-SAT to Independent Set

- We'll transform a 3-CNF formula into a graph that has an independent set of a certain size if and only if the formula is satisfiable.
- The graph has one vertex for each instance of each literal in the formula.
- Two nodes are connected by an edge if (1) they correspond to literals in the same clause, or (2) they correspond to a variable and its inverse.

Reduction from 3-SAT to Independent Set

$$(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$$



A graph derived from a 3CNF formula, and an independent set of size 4.
Black edges join literals from the same clause; red (heavier) edges join contradictory literals.

Reduction from 3-SAT to Independent Set

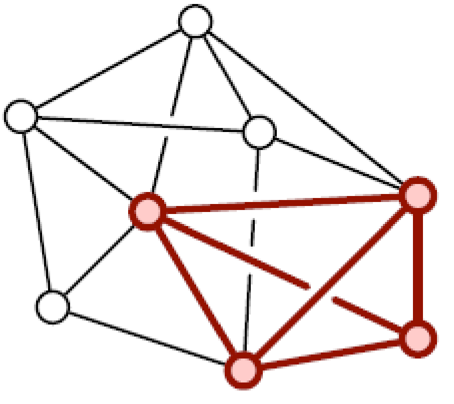
- Suppose that the original formula had *k* clauses. Then, I claim that the formula is satisfiable if and only if the graph has an independent set of size *k*.
- 1) independent set => satisfying assignment : If the graph has an independent set of *k* vertices, then each vertex must come from a different clause. Assign TRUE to each literal in the independent set. This assignment is consistent since contradictory literals are connected by edges. Assign any value to variables that have no literal in the independent set. The resulting assignment satisfies the original 3-CNF formula.
- 2) satisfying assignment => independent set : If we have a satisfying assignment, then we can choose one literal in each clause that is TRUE. Those literals form an independent set in the graph.

Independent Set is NP-Complete

- The reduction from 3-CNF formula to graph takes polynomial time and the graph size is also O(size of 3-CNF formula). Thus, IND-SET is NP-hard.
- IND-SET is NP.
  - If a graph has an independent set of size *k*, then the independent set is the certificate. We can check if there is any pair among the set is adjacent in polynomial time.

<u>Clique problem</u>

- A *clique* in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E.
- A clique is a complete subgraph of G.
- The size of a clique is the number of vertices it contains.
- ***The maximum clique problem*** is the optimization problem of finding a clique of maximum size in a graph.
- A decision problem (CLIQUE) asks whether a clique of a given size k exists in the graph.



A graph with maximum clique size 4.

Clique problem

- Lemma : CLIQUE is NP-hard.
- Proof : reduction from 3-SAT (in CLRS textbook.)

Reduction from 3-SAT to Clique

For a formula with *k* clauses,

- Each clause is represented by 3 vertices.
- 2 vertices are connected by an edge if they do not belong to the same clause and they are not negations of each other.

- In a satisfying truth assignment, there is at least one true literal in each clause. The true literals form a clique.
- Conversely, a clique of *k* or more vertices covers all clauses and thus implies a satisfying truth assignment.

Reduction from 3-SAT to Clique

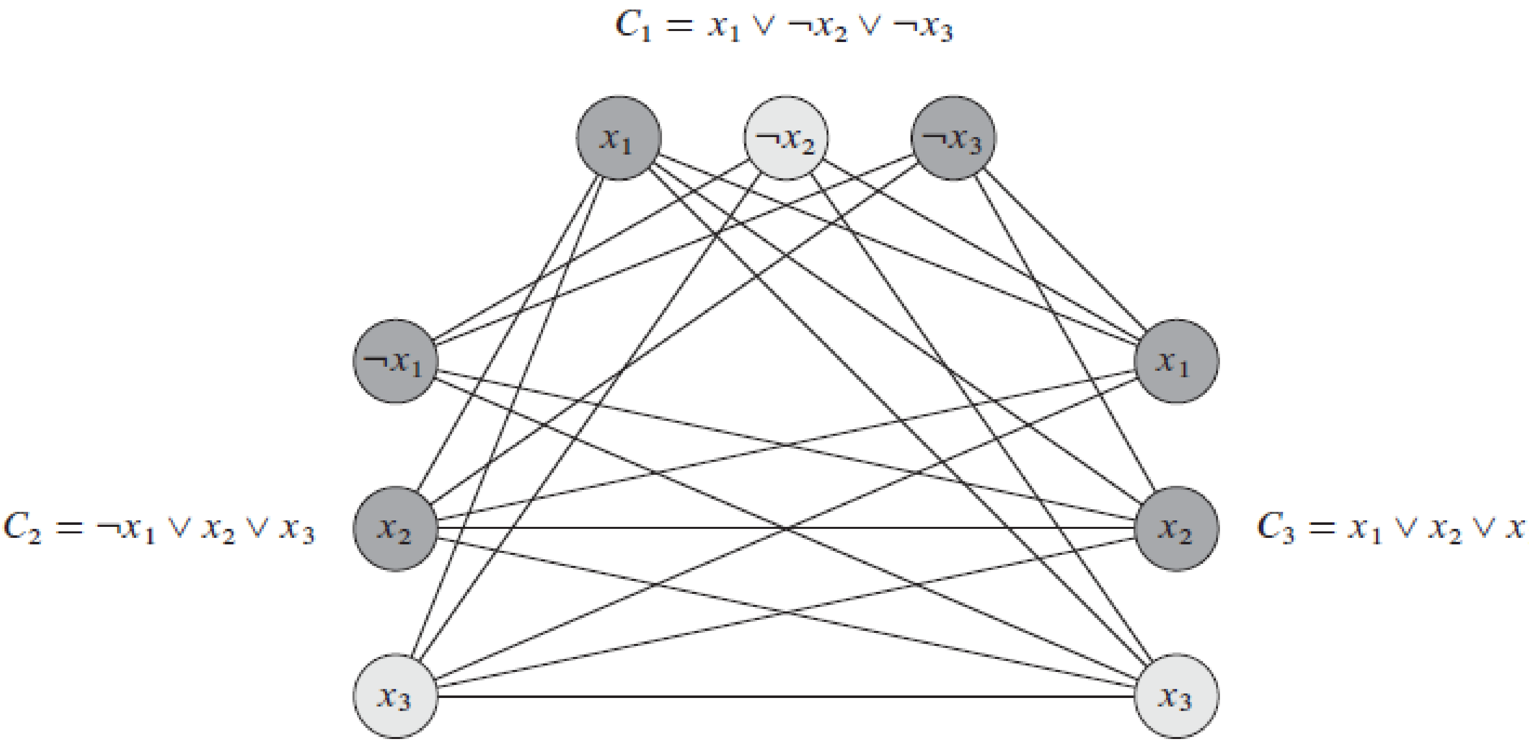
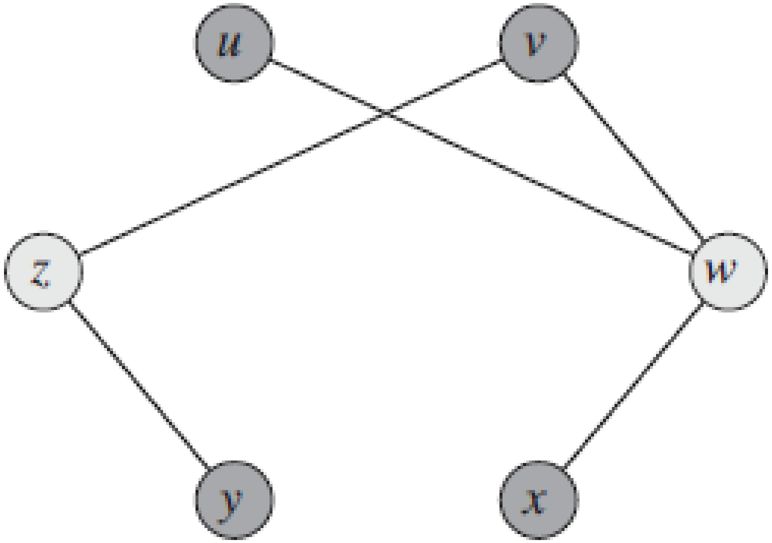


Figure 34.14 The graph G derived from the 3-CNF formula $\phi = C_1 \wedge C_2 \wedge C_3$, where $C_1 = (x_1 \vee \neg x_2 \vee \neg x_3)$, $C_2 = (\neg x_1 \vee x_2 \vee x_3)$, and $C_3 = (x_1 \vee x_2 \vee x_3)$, in reducing 3-CNF-SAT to CLIQUE. A satisfying assignment of the formula has $x_2 = 0$, $x_3 = 1$, 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

- Given a graph $G = (V, E)$, a subset $V' \subseteq V$ is a ***vertex cover*** if every edge has at least one endpoint in V'.
- VERTEX-COVER problem asks whether there is a vertex cover of a given size k.
- VERTEX-COVER $\in$ NPC. (reduction from CLIQUE)



Reduction from Clique to Vertex Cover

G has a clique of size k iff the complement of G has a vertex cover of size $|V|$ - k.

- Suppose G has a clique $V' \subseteq V$ with $|V'| = k$.
- Let (u,v) be any edge in the complement of G. $\rightarrow$ at least one of u or v does not belong to V'.
- So $u \in V - V'$ or $v \in V - V' \rightarrow$ edge (u,v) is covered by V-V'.
- Thus, V-V' forms a vertex cover of the complement of G with size $|V|$-k.

- Conversely, suppose that the complement of G has a vertex cover $V' \subseteq V$, where $|V'| = |V| - k$.
- Then, for all $u, v \in V$, if (u, v) is an edge in the complement of G, then $u \in V'$ or $v \in V'$ or both.
- The contrapositive is that for all $u, v \in V$, if $u \notin V'$ and $v \notin V'$, then $(u, v) \in E$.
- Thus, V-V' is a clique and it has size $|V| - |V'| = k$.

Reduction from Clique to Vertex Cover

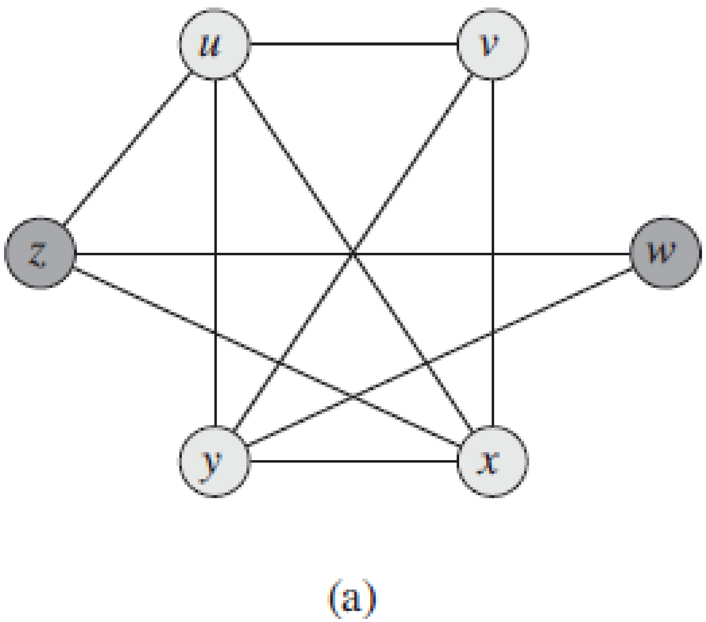
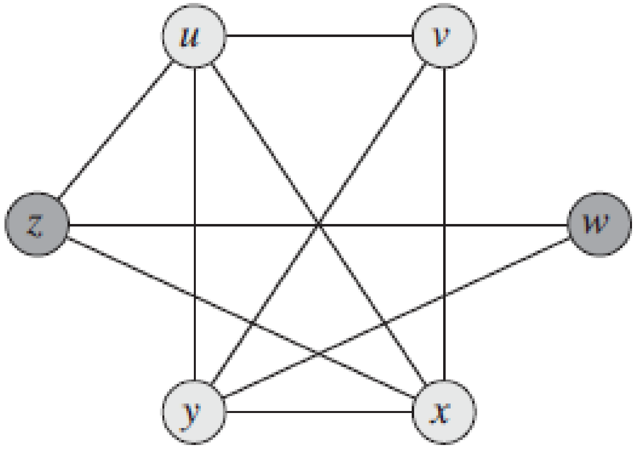
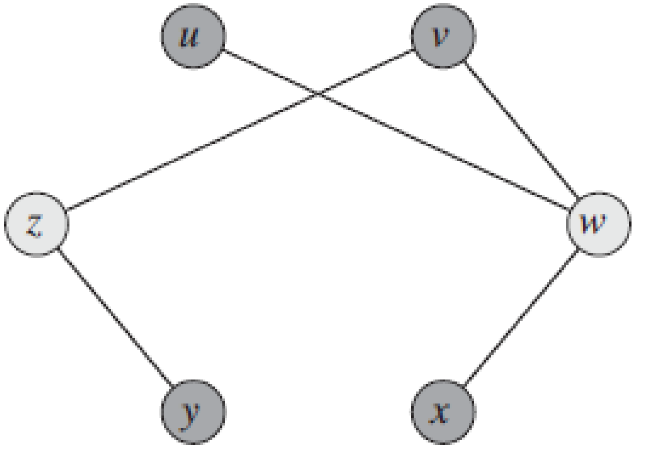


Figure 34.15 Reducing CLIQUE to VERTEX-COVER. **(a)** An undirected graph $G = (V, E)$ with clique $V' = \{u, v, x, y\}$. **(b)** The graph $\overline{G}$ produced by the reduction algorithm that has vertex cover $V - V' = \{w, z\}$.

Reduction from Clique to Independent Set

- $W \subseteq V$ is independent iff W defines a clique in the complement graph.
- To prove CLIQUE $\leq_p$ IND-SET, transform an instance H, k of the CLIQUE problem to the instance G = the complement of H, k of the IND-SET problem.
- G has an independent set of size k or larger iff H has a clique of size k or larger.





Theorem : the following are equivalent for $G = (V, E)$ and subset V' of V.

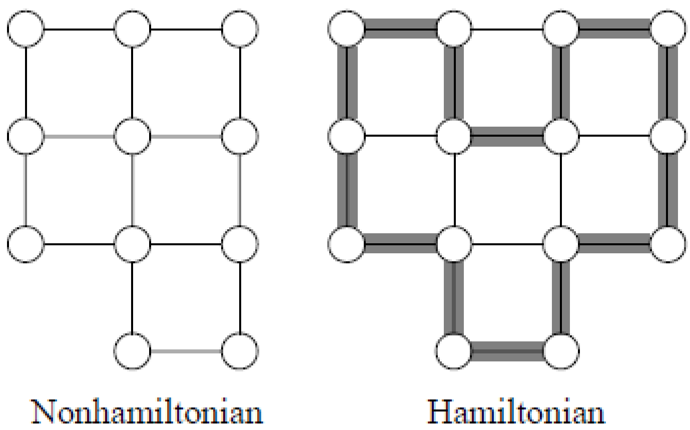
(1) V' is a clique of G.

(2) V' is an independent set of the complement of G.

(3) $V-V'$ is a vertex cover of the complement of G.

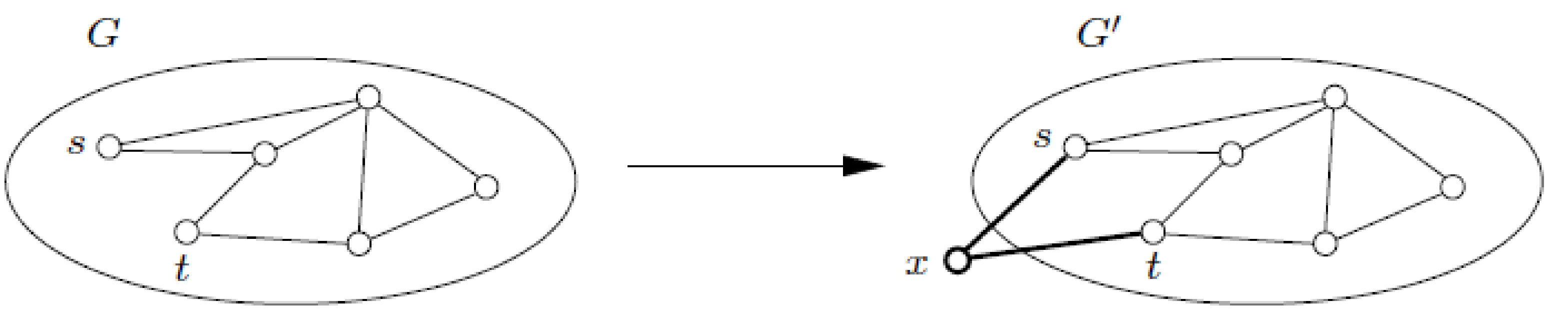
Hamiltonian cycles

- A hamiltonian cycle in a graph is a cycle that visits every vertex exactly once.
- (In textbook by DPV, it's called RUDRATA cycle.)
- The graph *G* is hamiltonian if it has a hamiltonian cycle.
- ***The hamiltonian cycle problem*** (HAM) asks whether a given graph is hamiltonian.



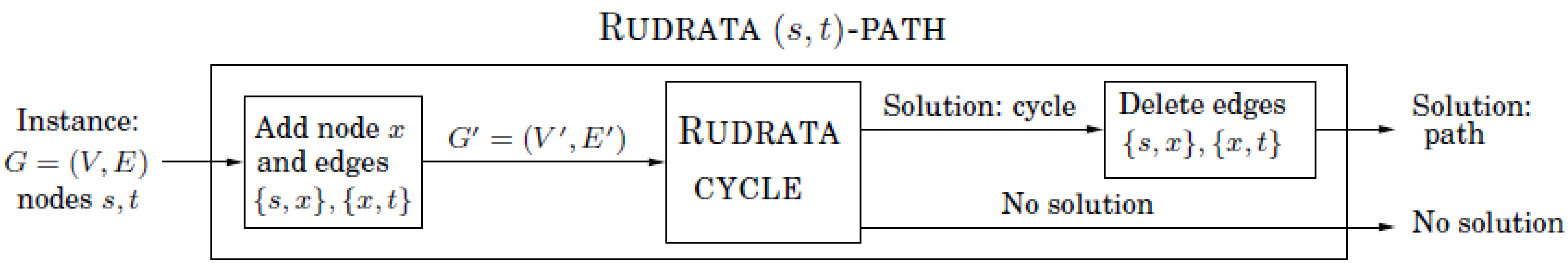
Hamiltonian path

- Given a graph G and two vertices s, t, the ***hamiltonian path*** problem asks whether G has a path from s to t that goes every vertex exactly once.
- (In textbook by DPV, it's called RUDRATA (s,t)-path problem)
- We can reduce the hamiltonian path problem to the hamiltonian cycle problem.
- Map an instance $G = (V, E)$, s, t of the hamiltonian path problem to an instance $G' = (V', E')$ of the hamiltonian cycle problem where $V' = V \cup \{x\}$ and $E' = E \cup \{\{s, x\}, \{x, t\}\}$.



Reduction from hamiltonian path to hamiltonian cycle

- *G'* has a Hamiltonian cycle if and only if *G* has a hamiltonian path from *s* to *t*.
- => Given a hamiltonian cycle, we get a hamiltonian path from *s* to *t* by deleting edges {*s*, *x*}, {*x*, *t*}.
- <= Given a hamiltonian path from *s* to *t* in *G*, we get a hamiltonian cycle in *G'* by adding edges {*s*, *x*}, {*x*, *t*}.



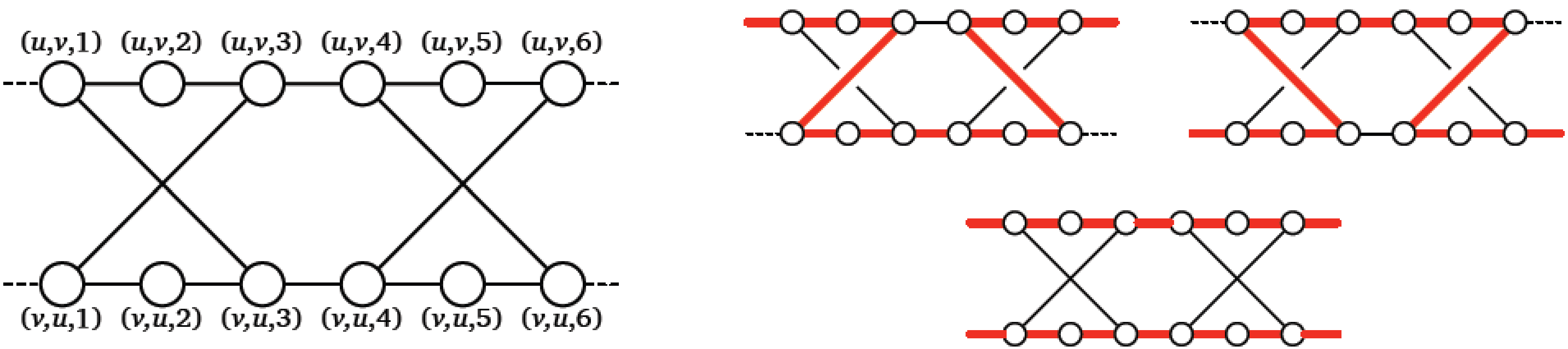
- We can also reduce hamiltonian cycle problem to hamiltonian path problem. (how?)

Hamiltonian cycle is NP-complete

- ***The hamiltonian cycle problem*** (HAM) asks whether a given graph is hamiltonian.
- HAM is NP-complete.
  - HAM is NP.
    - The certificate : the sequence of $|V|$ vertices that make up the hamiltonian cycle.
    - We can check whether the sequence contains each vertex in V once and forms a cycle in polynomial time.
  - HAM is NP-hard.
    - reduction from VERTEX COVER

Reduction from Vertex Cover to HAM

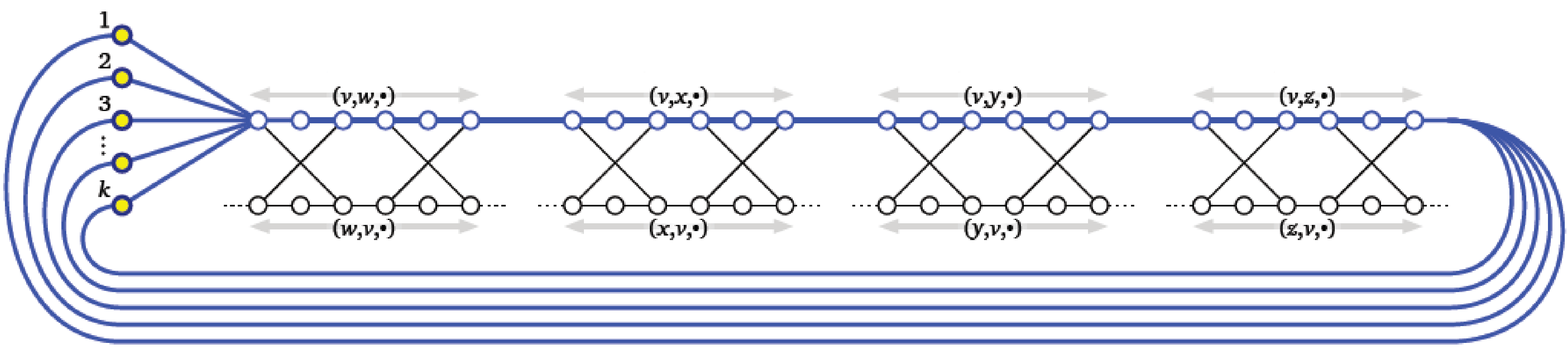
- Given a graph *G* and an integer *k*, we need to transform it into another graph *G'*, such that *G'* has a hamiltonian cycle if and only if *G* has a vertex cover of size *k*.
- For each edge (*u*, *v*) in *G*, we have an *edge gadget* in *G'* consisting of 12 vertices and 14 edges. 4 corner vertices (*u*, *v*, 1), (*u*, *v*, 6), (*v*, *u*, 1), (*v*, *u*, 6) have an edge leaving the gadget. A hamiltonian cycle can only pass through an edge gadget in only 3 ways.



An edge gadget for (u, v) and the only possible Hamiltonian paths through it.

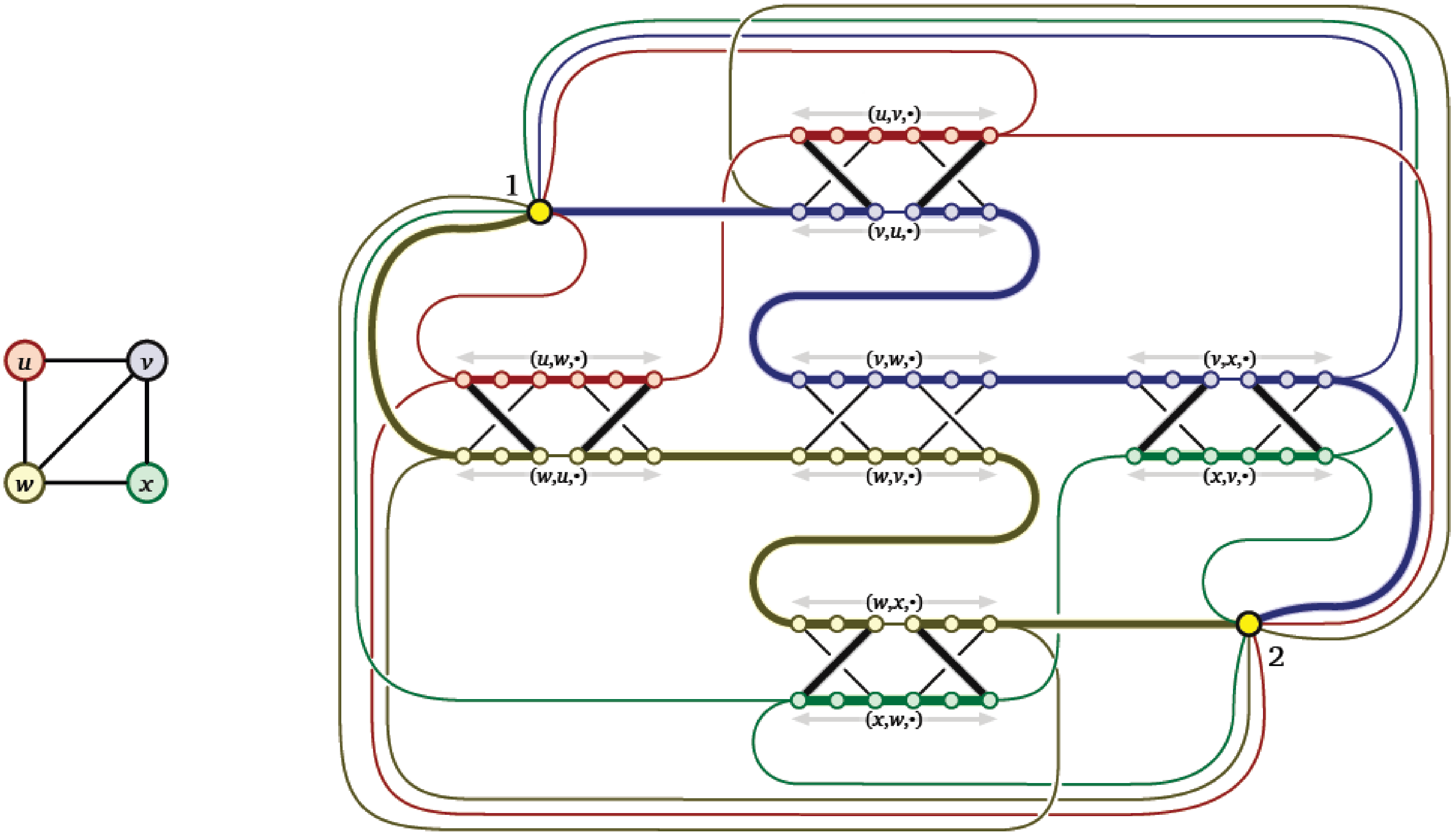
Reduction from Vertex Cover to HAM

- *G'* also contains *k cover vertices*, simply numbered 1 through *k*.
- For each vertex u in G, we string together all the edge gadgets for edges (u, v) into a single *vertex chain*, and then connect the edges of the chain to all the cover vertices.
- Suppose vertex u has d neighbors $v_1, v_2, \ldots, v_d$. Then, G' has d-1 edges between $(u, v_i, 6)$ and $(u, v_{i+1}, 1)$, plus k edges between the cover vertices and $(u, v_1, 1)$, and finally k edges between the cover vertices and $(u, v_d, 6)$.



The vertex chain for v: all edge gadgets involving v are strung together and joined to the k cover vertices.

Reduction from Vertex Cover to HAM



The original graph G with vertex cover $\{v, w\}$, and the transformed graph G' with a corresponding Hamiltonian cycle. Vertex chains are colored to match their corresponding vertices.

Reduction from Vertex Cover to HAM

- G has a vertex cover of size k if and only if G' has a hamiltonian cycle.

- => If $\{v_1, v_2, \ldots, v_k\}$ is a vertex cover of G, then G' has a hamiltonian cycle.
- Start at cover vertex 1, traverse the vertex chain for v_1 , then visit cover vertex 2, then traverse the vertex chain for v_2, and so forth, return to cover vertex 1.

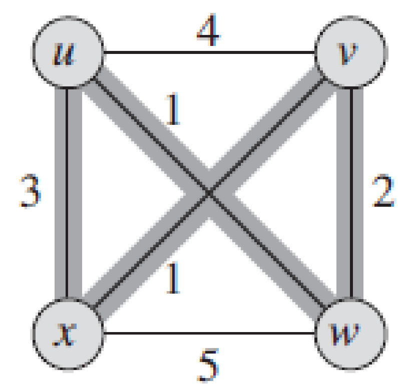
- <= Conversely, if G' has a hamiltonian cycle C in G', G has a vertex cover of size k.
- C alternates between cover vertices and vertex chains, and the vertex chains correspond to the k vertices in a vertex cover of G.

Reduction from Vertex Cover to HAM

- The size of *G'* is polynomial in the size of *G*, and hence we can construct *G'* in polynomial time in the size of *G* (in $O(n^2)$ time.)
- Thus, HAM is NP-hard.

Traveling salesman

- Assume a complete graph.
- Each edge has a non-negative integer weight.
- The traveling salesman problem (TSP) asks whether there is a permutation of the vertices s.t. the sum of edges connecting contiguous vertices (and the last vertex to the first) is k or less.
- TSP is NP-complete.
- TSP is NP because we can use the sequence of n vertices in the tour as a certificate.
  - The verification algorithm checks that this sequence contains each vertex exactly once, sums up the edge costs, and checks whether the sum is at most k, which can be done in polynomial time.
- TSP is NP-hard (use reduction from HAM.)

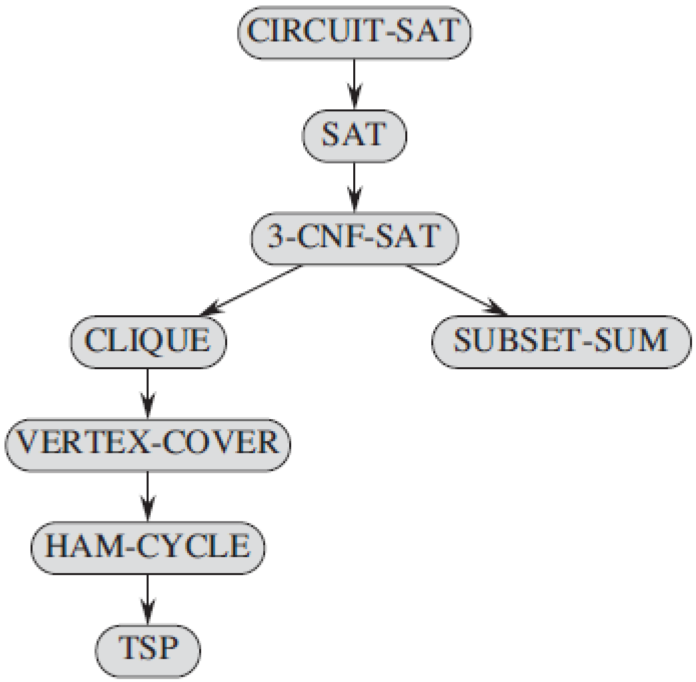


Reduction from HAM to TSP

- For a graph G in hamiltonian cycle problem, construct an instance of TSP by forming the complete graph $G' = (V, E')$, where $E' = \{(i, j) : i, j \in V \text{ and } i \neq j\}$. If there is an edge between two vertices in G, assign 0 as edge weight, otherwise, 1.
- Show that G has a hamiltonian cycle h if and only if G' has a TSP tour with weight 0.
- => Suppose that G has a hamiltonian cycle h. Then, each edge in h has weight 0 in G'. Thus, h is a tour in G' with weight 0.
- <= Conversely, suppose that G' has a TSP tour h' of cost at most 0. Since the costs of the edges in E' are 0 and 1, the cost of h' is 0 and each edge on the tour must have cost 0. Therefore, h' contains only edges in E. Thus, h' is a hamiltonian cycle in G.

NP-hard problems

- CIRCUIT-SAT, SAT, 3-SAT
- IND-SET, CLIQUE, VERTEX-COVER
- HAM, TSP
- SUBSET-SUM : Given a finite set *S* of positive integers and an integer target $t > 0$, is there a subset $S' \subseteq S$ whose element sum to *t*?
- 3-COLOR : Given a graph, can it be 3-colored? (2-COLOR $\in$ P)
- *Minesweeper*, *Tetris*, *Sudoku* are also shown to be NP-hard.



Hierarchy of complexity classes

- PSACE : the set of decision problems that can be solved using *polynomial space.*
- EXP : the set of decision problems that can be solved in *exponential time*.
- NEXP : the class of decision problems that can be solved in *nondeterministic exponential time;* equivalently, a decision problem is in NEXP if and only if, for every YES instance, there is a certificate of this fact that can be checked in exponential time.
- EXPSPACE : the set of decision problems that can be solved using *exponential space.*
- P $\subseteq$ NP $\subseteq$ PSPACE $\subseteq$ EXP $\subseteq$ NEXP $\subseteq$ EXPSPACE ……