Review session: Exam question 5
NP complete problems.

- Lecture 19
  - NP-completeness of variants of SAT: 3SAT, 2SAT, and MAX2SAT.
  - Papadimitriou, pages 183-187.
- Lecture 20 (one hour)
  - NP-completeness of NAESAT, 3-COLORING, and MAXCUT.
  - Papadimitriou, pages 187-188, 198 (mid)-199 (mid).
- Lecture 21
  - NP-completeness of INDEPENDENT SET (+CLIQUE & VERTEX COVER), HAMILTON PATH, and TSP.
  - Papadimitriou, pages 188(mid)-191, 193(bottom)-198 (mid).
- Lecture 22 (one hour)
  - NP-completeness of TRIPARTITE MATCHING, EXACT COVER BY 3-SETS, SET COVER, and SET PACKING.
  - Papadimitriou, pages pages 199(mid)-201.
- Lecture 23
  - NP-completeness of KNAPSACK and BIN PACKING. Strong NP-completeness and the use of NP completeness.
  - Papadimitriou, pages 202-206.

How to establish NP-hardness

Lemma

If L_1 is NP-hard and $L_1 \leq L_2$, then L_2 is NP-hard

SAT and relatives

SAT NP-complete

- Given: CNF formula F on n variables.
- Question: Does there exist $x \in \{0,1\}^n$ such that $F(x) = 1$?

CircuitSAT NP-complete

- Given: Boolean Circuit C on n variables.
- Question: Does there exist $x \in \{0,1\}^n$ such that C$(x) = 1$?

kSAT NP-complete for $k = 3$ and in P for $k = 2$.

- Given: kCNF formula F on n variables.
- Question: Does there exist $x \in \{0,1\}^n$ such that $F(x) = 1$?

NAESAT NP-complete

- Given: 3CNF formula F on n variables.
- Question: Does there exist $x \in \{0,1\}^n$ such that in every clause, all of the literals are *not* the same?
  (i.e. all clauses should be satisfied as usual, but their literals are *not* allowed to all be true).

3SAT is NP-complete

Recall reduction CircuitSAT $\leq$ SAT (Tseitin transformation):

Let C be a Boolean circuit on n variables.

Construct CNF formula F with

- Variables: One variable g for every gate g of C.
- Clauses:
  - For each gate of C, clauses that express the computation of the gate. E.g., $g \Leftrightarrow h_1 \wedge h_2$ expresses that gate g is the Boolean conjunction of gates h_1 and h_2. For every gate this is a Boolean function on at most 3 variables, which can be expressed as a CNF formula.
  $$(\neg g \vee h_1) \wedge (\neg g \vee h_2) \wedge (g \vee \neg h_1 \vee \neg h_2)$$
  - For the output gate g_{out} of C, the unit clause (g_{out}).

NAESAT is NP-complete

Recall reduction CircuitSAT $\leq$ SAT (Tseitin transformation):

Let C be a Boolean circuit on n variables.

Construct CNF formula F with

- Variables: One variable g for every gate g of C.
- Clauses:
  - For each gate of C, clauses that express the computation of the gate. E.g., $g \Leftrightarrow h_1 \wedge h_2$ expresses that gate g is the Boolean conjunction of gates h_1 and h_2. For every gate this is a Boolean function on at most 3 variables, which can be expressed as a CNF formula.
    $$(\neg g \vee h_1) \wedge (\neg g \vee h_2) \wedge (g \vee \neg h_1 \vee \neg h_2)$$
  - For the output gate g_{out} of C, the unit clause (g_{out}).

Modification: Add a new (global) variable z to all length 1 and length 2 clauses.

Resolution

Let F be the following CNF formula:

$$(x \vee P_1) \wedge (x \vee P_2) \wedge \cdots \wedge (x \vee P_k) \wedge$$
$$(\neg x \vee Q_1) \wedge (\neg x \vee Q_2) \wedge \cdots \wedge (\neg x \vee Q_\ell) \wedge$$
$$R$$

where R does not contain the variable x.

Then $\mathrm{Resolve}(F, x)$ is defined to be the following CNF formula:

$$(P_1 \vee Q_1) \wedge (P_2 \vee Q_1) \wedge \cdots \wedge (P_k \vee Q_1) \wedge$$
$$(P_1 \vee Q_2) \wedge (P_2 \vee Q_2) \wedge \cdots \wedge (P_k \vee Q_2) \wedge$$
$$\ldots$$
$$(P_1 \vee Q_\ell) \wedge (P_2 \vee Q_\ell) \wedge \cdots \wedge (P_k \vee Q_\ell) \wedge$$
$$R$$

The Davis-Putnam procedure

Input: CNF formula F

Output: Satisfiability of F

while F is not empty **do**:

    if F contains an empty clause **then**:

        return false

    let $x \in \mathrm{vars}(F)$

    let $F \coloneqq \mathrm{resolve}(F, x)$

return true

The implication graph G(F) of a 2CNF F and the relation to 2SAT

$$(x \lor y) \equiv (\neg x \Rightarrow y) \equiv (x \Leftarrow \neg y)$$

Theorem

F is satisfiable if and only if:

there is no variable x such that there is *both* a path from x to $\neg x$ *and* from $\neg x$ to x in $G(F)$

$$(x_1 \lor x_2) \land (x_1 \lor \neg x_3) \land (\neg x_1 \lor x_2) \land (x_2 \lor x_3)$$

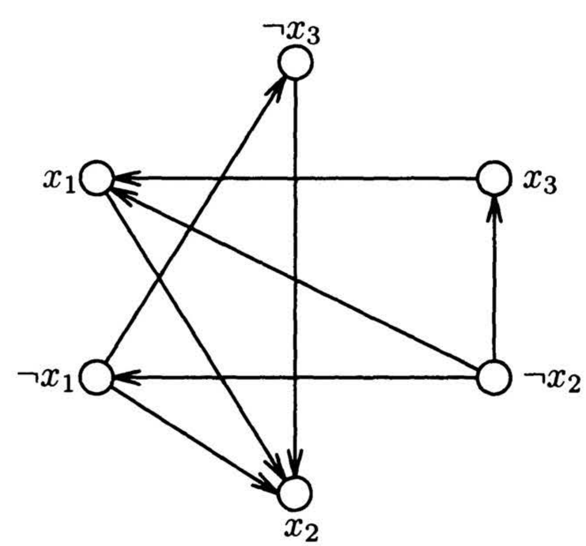


Figure 9-1. The algorithm for 2SAT.

Optimization version of SAT

MAX2SAT

- Given: 2CNF formula F on n variables. Target K.
- Question: Does there exist $x \in \{0,1\}^n$ satisfying at least K clauses of F?

Gadget:

$$(x) \wedge (y) \wedge (z) \wedge (w) \wedge$$
$$(\neg x \vee \neg y) \wedge (\neg y \vee \neg z) \wedge (\neg z \vee \neg x) \wedge$$
$$(x \vee \neg w) \wedge (y \vee \neg w) \wedge (z \vee \neg w)$$

3-COLORING

3-COLORING

- Given: Undirected graph $G = (V, E)$.
- Question: Does there exist a valid 3-coloring of G?
(i.e. a function $c: V \to \{0,1,2\}$ such that $c(u) \neq c(v)$ for all $uv \in E$)

Theorem 3-COLORING is NP-complete.

- Hardness: NAESAT $\leq$ 3-COLORING

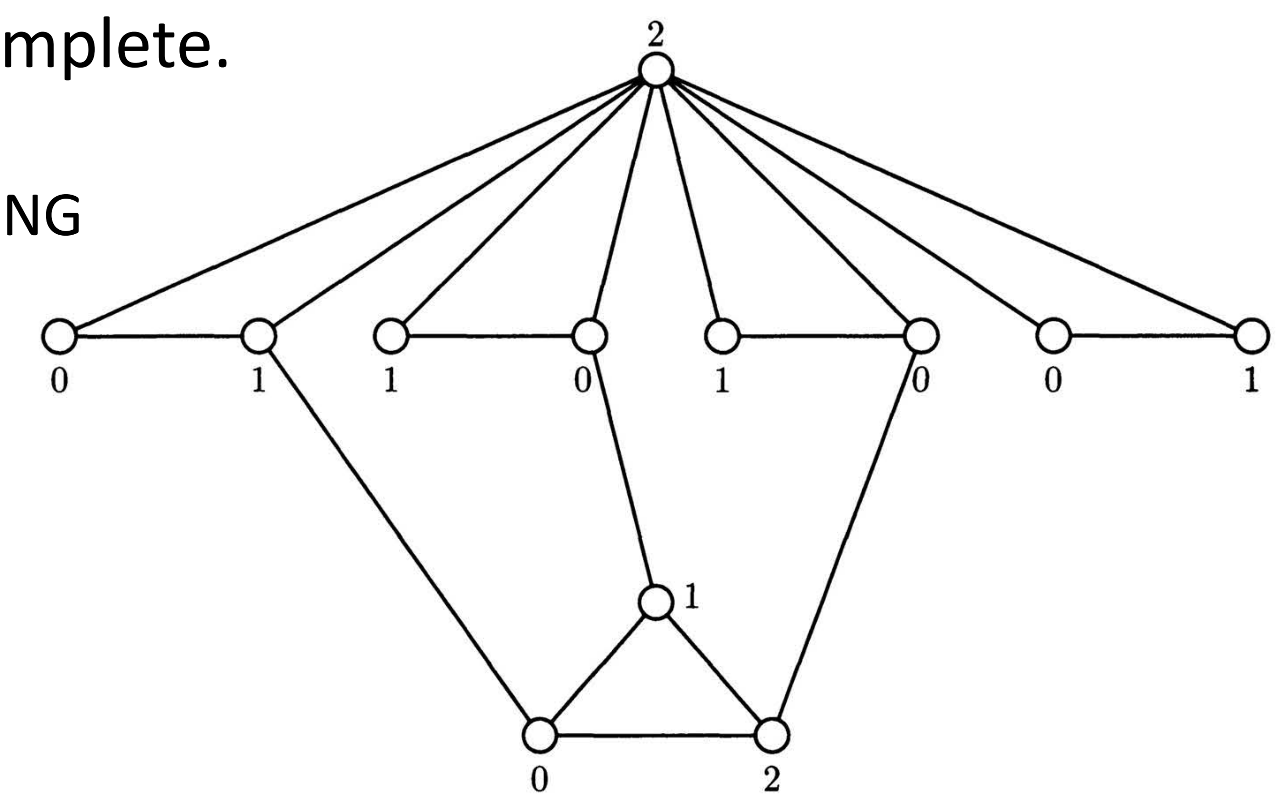


Figure 9-8. The reduction to 3-COLORING.

MAXCUT

MAX CUT

- Given: Undirected graph $G = (V, E)$, target K.
- Question: Does there exist a partition $V = S \cup T$ of G, such that the number $|E(S,T)|$ of edges between vertices of S and T is at least K?

Theorem MAX CUT is NP-complete.

- Hardness: NAESAT $\leq$ MAX CUT

(Note: The proof given in Papadimitriou is unnecessarily complicated). Take just one edge between literal pairs and use target value $K = n + 2m$

$$(x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \equiv$$
$$(x_1 \vee x_2 \vee x_2) \wedge (x_1 \vee \neg x_3 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$$

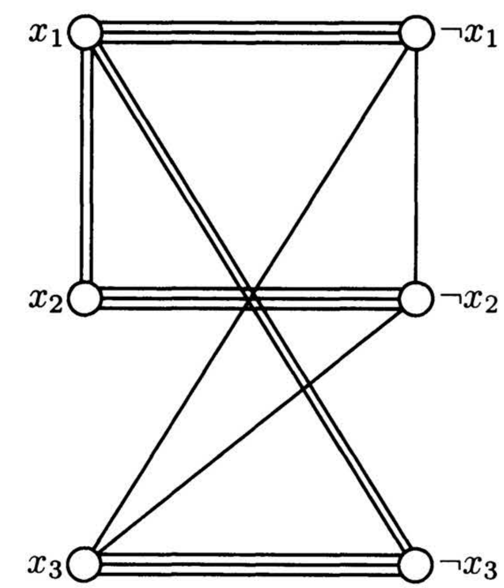


Figure 9-3. Reduction to MAX CUT.

3SAT: One of the most useful NP-hard problems for reductions

3SAT

- Given: 3CNF formula F on n variables.
- Question: Does there exist $x \in \{0,1\}^n$ such that $F(x) = 1$?

Theorem 3SAT is NP-complete.

Lemma

If L_1 is NP-hard and $L_1 \leq L_2$, then L_2 is NP-hard

When constructing a reduction 3SAT $\leq L$, our task is to figure our how to model truth assignments in the search space of L, and then how to express that clauses are satisfied: a “programming task”.

INDEPENDENT SET

INDEPENDENT SET

- Given: Undirected graph $G = (V, E)$, target K.
- Question: Does there exist an independent set I in G with $|I| \geq K$?

Theorem INDEPENDET SET is NP-complete.

- Hardness: 3SAT $\leq$ INDEPENDENT SET

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

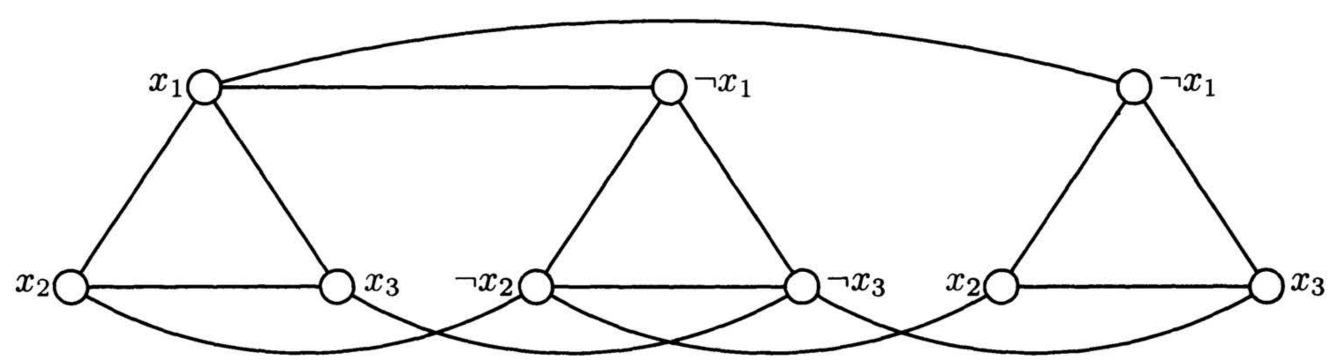


Figure 9-2. Reduction to INDEPENDENT SET.

Independent sets, cliques, and vertex covers

CLIQUE

- Given: Undirected graph $G = (V, E)$, target K.
- Question: Does there exist a clique Q in G with $|Q| \geq K$?

VERTEX COVER

- Given: Undirected graph $G = (V, E)$, budget B.
- Question: Does there a exist vertex cover C in G with $|C| \leq B$?

Observation:

I is independent set in $G \Leftrightarrow I$ is clique in $\bar{G} \Leftrightarrow V \setminus I$ is vertex cover in G.

Corollary CLIQUE and VERTEX COVER are both NP-complete.

HAMILTON PATH

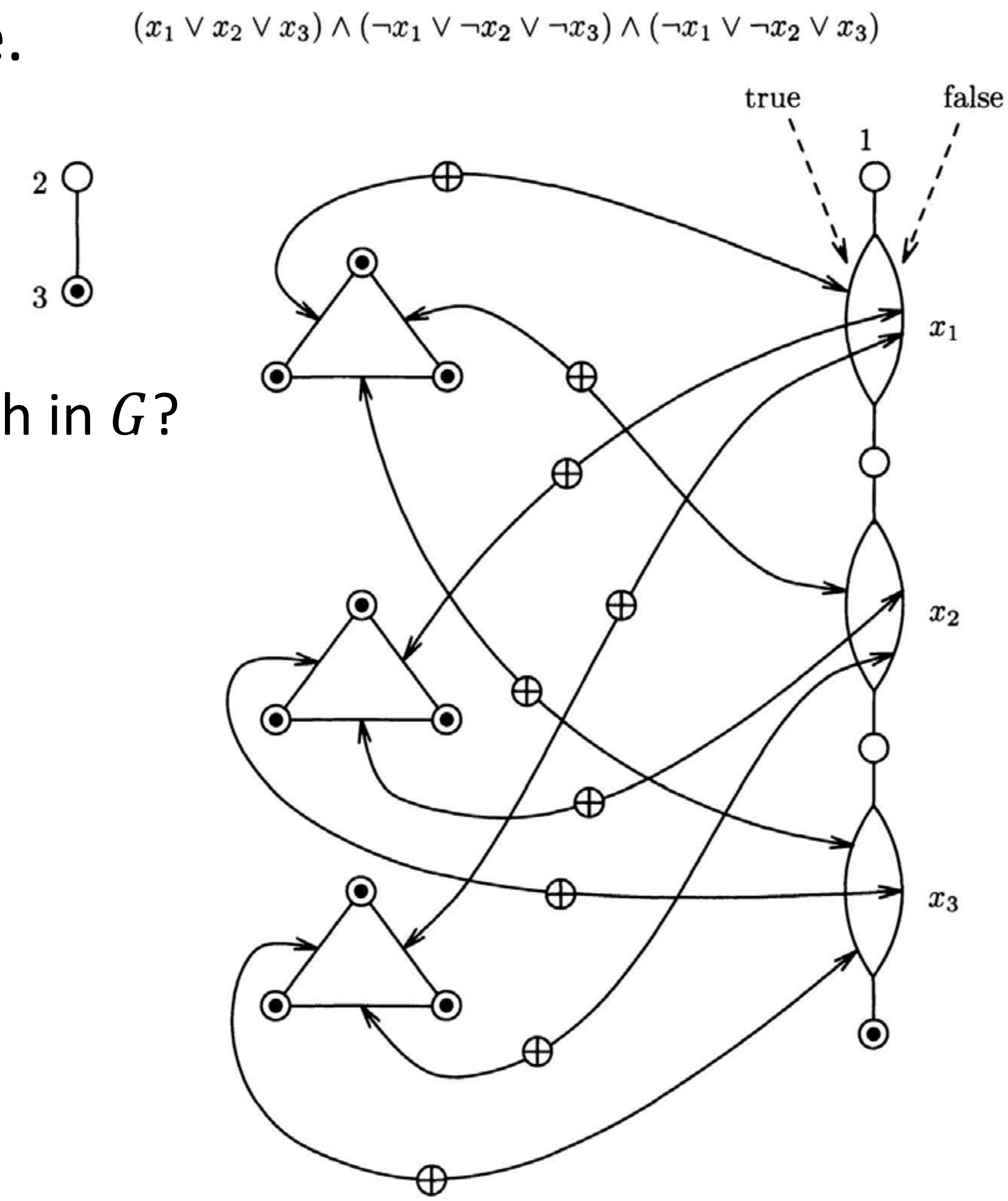
Let $G = (V, E)$ be an undirected graph. A Hamiltonian path in G is a path in G that visits each node exactly once.

HAMILTON PATH

- Given: Undirected graph $G = (V, E)$.
- Question: Does there exist a Hamiltonian path in G?

Theorem HAMILTON PATH is NP-complete.

- Hardness: 3SAT $\leq$ HAMILTON PATH



⊙ all these nodes are connected in a big clique.

HAMILTON PATH: Gadgets

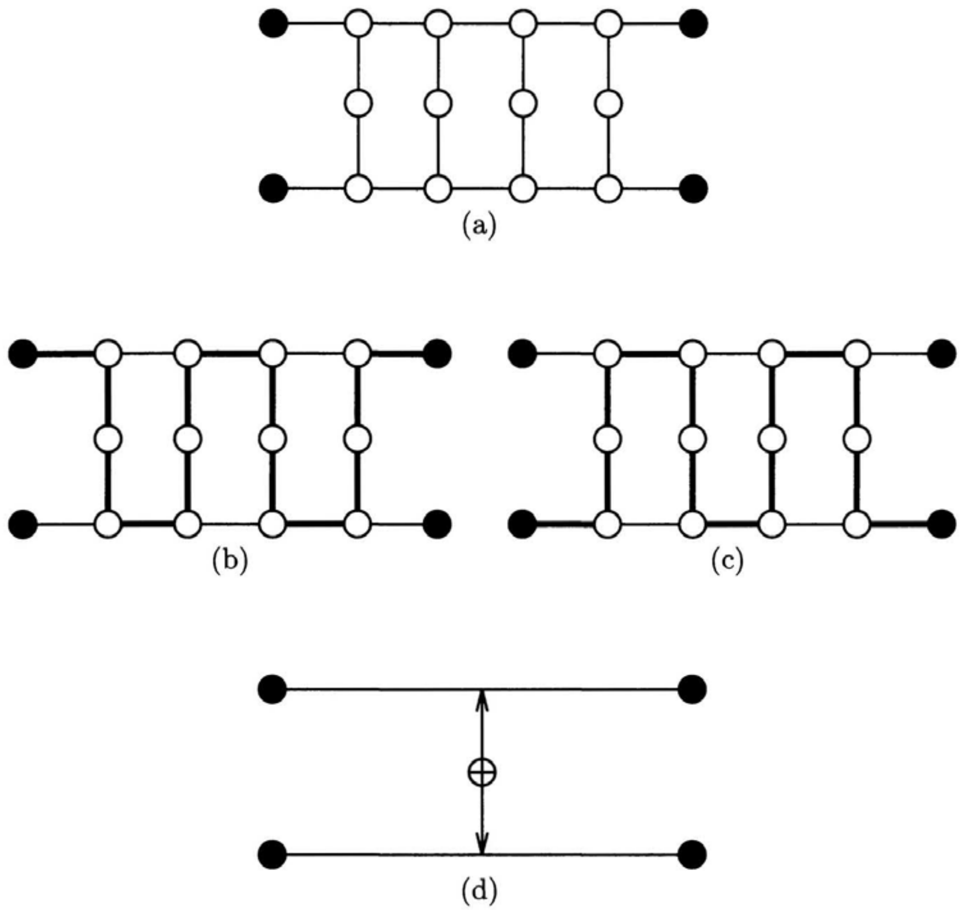


Figure 9-5. The consistency gadget.

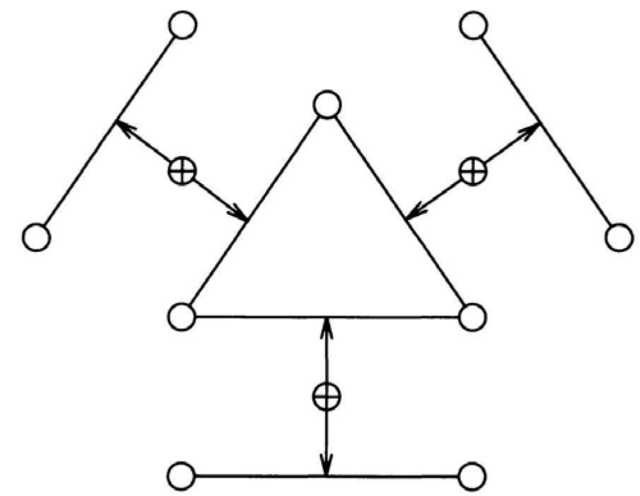


Figure 9-6. The constraint gadget.

The Travelling Salesman Problem

TSP

- Given: An $n \times n$ distance matrix $\mathrm{D} = [d_{ij}]$, budget B.
- Question: Does there exist a permutation π on $\{0, \ldots, n-1\}$ such that $\sum_{i=0}^{n-1} d_{\pi(i),(\pi((i+1) \bmod n)} \leq B$?

Theorem TSP is NP-complete.

- Hardness: HAMILTON PATH ≤ TSP

Given graph $G = (V, E)$, $n = |V|$.

$$d_{ij} = \begin{cases} 1 & \text{if } ij \in E \\ 2 & \text{if } ij \notin E \end{cases}$$

$$B = n + 1$$

TRIPARTITE MATCHING
(a.k.a. 3-dimensional matching)

TRIPARTITE MATCHING

- Given: Sets B, G, H with $|B| = |G| = |H| = n$ and triples $T \subseteq B \times G \times H$.
- Question: Does there exist $M \subseteq T$ such that $|M| = n$ and for every pair of triples $(b, g, h), (b', g', h') \in M$ we have $(b \neq b') \wedge (g \neq g') \wedge (h \neq h')$?

TRIPARTITE MATHCING is the generalization of BIPARTITE MATCHING asking about existence of a perfect matching in a bipartite graph, to asking about existence of a perfect matching in a *tripartite 3-uniform hypergraph*.

Think:

- B = set of n boys.
- G = set of n girls.
- H = set of n homes.

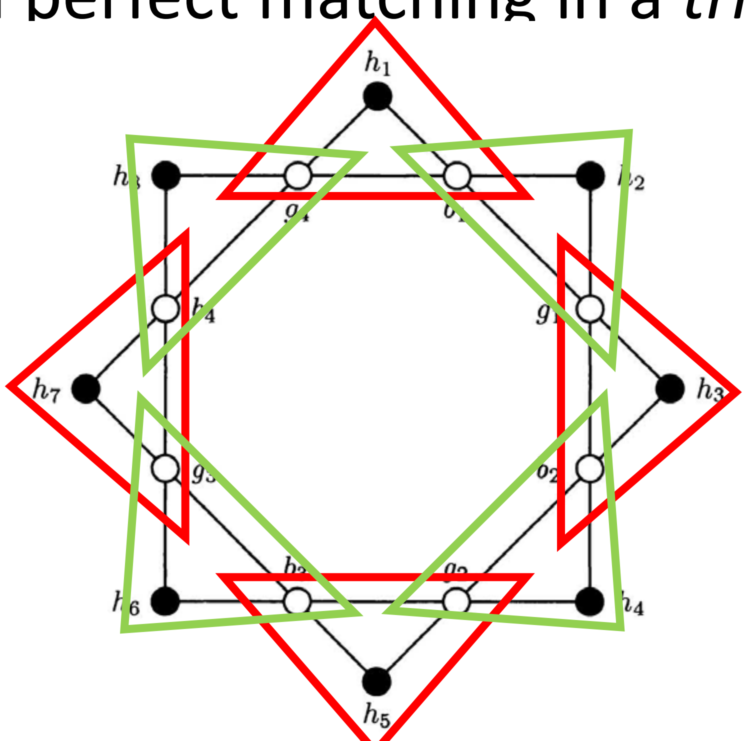


Figure 9-9. The choice-consistency gadget.

Warning: Not a graph!
(but a hypergraph)

SET COVER and friends

EXACT COVER BY 3-SETS

- Given: Finite set U of size n, and set $\mathcal{F}$ of subsets of U such that $\bigcup_{S\in\mathcal{F}} S = U$ and $|S| = 3$ for all $S \in \mathcal{F}$.
- Question: Does there exist $\mathcal{C} \subseteq \mathcal{F}$ such that $\bigcup_{S\in\mathcal{C}} S = U$ and $3|\mathcal{C}| = n$?

SET COVER

- Given: Finite set U of size n, set $\mathcal{F}$ of subsets of U, and budget B.
- Question: Does there exist $\mathcal{C} \subseteq \mathcal{F}$ such that $\bigcup_{S\in\mathcal{C}} S = U$, and $|\mathcal{C}| \leq B$?

SET PACKING

- Given: Finite set U of size n, set $\mathcal{F}$ of subsets of U, and target K.
- Question: Does there exist $\mathcal{C} \subseteq \mathcal{F}$ such that $S \cap S' = \emptyset$, for all S, $S' \in \mathcal{C}$ with $S \neq S'$, and $|\mathcal{C}| \geq$K?

Corollary EXACT COVER BY 3-SETS, SET COVER, and SET PACKING are all NP-complete.

KNAPSACK

KNAPSACK

- Given: Weights $w_1, \dots, w_n$, values $v_1, \dots, v_n$, weight budget B, and target K.
- Question: Does there exist $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} w_i \leq B$ and $\sum_{i \in S} v_i \geq K$?

SUBSET SUM

- Given: Numbers $a_1, \dots, a_n$, and target K.
- Question: Does there exist $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} a_i = K$?

- Hardness: ECB3S $\leq$ SUBSET SUM

$$
\begin{array}{ccccccccccccc}
\rightarrow & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\rightarrow & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\rightarrow & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\rightarrow & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
+ & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

Figure 9.10. Reduction to KNAPSACK.

Binary vs. Unary

- $L_{KNAPSACK}^{binary}$ is NP-hard.

- KNAPSACK can be solved in time O(n W)
  (on a random access machine).

- Therefore $L_{KNAPSACK}^{unary}$ is in P:
  "Knapsack has a pseudopolynomial algorithm"

- Unless P=NP, any reduction of an NP-hard problem to KNAPSACK has to involve "large numbers".

Pseudopolynomial algorithms

- Let a problem Q involving integers be given.
- If L_Q^{unary} is in P, we say that Q has a *pseudopolynomial* algorithm.
- If L_Q^{unary} is NP-hard, we say that Q is *strongly NP-hard.*
- If a strongly NP-hard problem has a pseudopolynomial algorithm, then P=NP.

Bin packing

BIN PACKING

- Given: n positive integers $a_1, a_2, \ldots, an$ (items), a positive integer *B* (number of bins) and a positive integer *C* (the capacity of a bin).
- Question: Does there exist a way to place all items in the bins?

TRIPARTITE MATCHING can be reduced to BIN PACKING with all integers in the output being $O(n^4)$.

BIN PACKING is strongly NP-hard: A pseudopolynomial algorithm would imply P=NP.

Item	Size
first occurrence of a boy $b_i[1]$	$10M^4 + iM + 1$
other occurrences of a boy $b_i[q], q > 1$	$11M^4 + iM + 1$
first occurrence of a girl $g_j[1]$	$10M^4 + jM^2 + 2$
other occurrences of a girl $g_j[q], q > 1$	$11M^4 + jM^2 + 2$
first occurrence of a home $h_k[1]$	$10M^4 + kM^3 + 4$
other occurrences of a home $h_k[q], q > 1$	$8M^4 + kM^3 + 4$
triple $(b_i, g_j, h_k) \in T$	$10M^4 + 8-$ $-iM - jM^2 - kM^3$

Figure 9.11. The items in BIN PACKING.

How to solve NP-hard problems

- <u>**Irony of NP-hardness**</u>: It is often ***easier*** to find the ***best*** way of solving an NP-hard problem than the ***best*** way of solving a problem in P, because ***you know which approaches (not) to try.***

- Algorithmic patterns for NP-hard problems: ***branch-and-bound, branch-and-cut, branch-and-reduce***.