You Have an NP-Complete Problem (for Your Thesis)

- From Propositions 27 (p. 242) and Proposition 30 (p. 245), it is the least likely to be in P.
- Your options are:
  - Approximations.
  - Special cases.
  - Average performance.
  - Randomized algorithms.
  - Exponential-time algorithms that work well in practice.
  - "Heuristics" (and pray that it works *for your thesis*).

I thought NP-completeness was an interesting idea:
I didn't quite realize its potential impact.
— Stephen Cook, in Shasha & Lazere (1998)

I was indeed surprised by Karp's work
since I did not expect so many
wonderful problems were NP-complete.
— Leonid Levin, in Shasha & Lazere (1998)

3SAT

- k-SAT, where $k \in \mathbb{Z}^+$, is the special case of SAT.
- The formula is in CNF and all clauses have *exactly* k literals (repetition of literals is allowed).
- For example,

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

3SAT Is NP-Complete

- Recall Cook's Theorem (p. 265) and the reduction of CIRCUIT SAT to SAT (p. 231).
- The resulting CNF has at most 3 literals for each clause.
  - This accidentally shows that 3SAT where each clause has at most 3 literals is NP-complete.
- Finally, duplicate one literal once or twice to make it a 3SAT formula.
- Note: The overall reduction remains parsimonious.

The Satisfiability of Random 3SAT Expressions

- Consider a random 3SAT expressions ϕ with n variables and cn clauses.
- Each clause is chosen independently and uniformly from the set of all possible clauses.
- Intuitively, the larger the c, the less likely ϕ is satisfiable as more constraints are added.
- Indeed, there is a c_n such that for $c < c_n(1-\epsilon)$, ϕ is satisfiable almost surely, and for $c > c_n(1+\epsilon)$, ϕ is unsatisfiable almost surely.[a]

[a] Friedgut and Bourgain (1999). As of 2006, $3.52 < c_n < 4.596$.

Another Variant of 3SAT

Proposition 36 3SAT *is NP-complete for expressions in which each variable is restricted to appear at most three times, and each literal at most twice. (*3SAT *here requires only that each clause has* at most *3 literals.)*

- Consider a general 3SAT expression in which x appears k times.
- Replace the first occurrence of x by x_1, the second by x_2, and so on, where $x_1, x_2, \ldots, x_k$ are k *new* variables.

The Proof (concluded)

- Add $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \cdots \wedge (\neg x_k \vee x_1)$ to the expression.
  - It is logically equivalent to

$$x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_k \Rightarrow x_1.$$

  - Note that each clause above has only 2 literals.
- The resulting equivalent expression satisfies the conditions for x.

An Example

- Suppose we are given the following 3SAT expression

$$\cdots (\neg x \vee w \vee g) \wedge \cdots \wedge (x \vee y \vee z) \cdots .$$

- The transformed expression is

$$\cdots (\neg x_1 \vee w \vee g) \wedge \cdots \wedge (x_2 \vee y \vee z) \cdots (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_1).$$

  - Variable x_1 appears 3 times.
  - Literal x_1 appears once.
  - Literal $\neg x_1$ appears 2 times.

2SAT Is in NL $\subseteq$ P

- Let ϕ be an instance of 2SAT: Each clause has 2 literals.
- NL is a subset of P (p. 201).
- By Eq. (2) on p. 211, coNL equals NL.
- We need to show only that recognizing unsatisfiable expressions is in NL.
- See the textbook for proof.

Generalized 2SAT: MAX2SAT

- Consider a 2SAT expression.
- Let $K \in \mathbb{N}$.
- MAX2SAT is the problem of whether there is a truth assignment that satisfies at least K of the clauses.
  - MAX2SAT becomes 2SAT when K equals the number of clauses.
- MAX2SAT is an optimization problem.
- MAX2SAT $\in$ NP: Guess a truth assignment and verify the count.
- We now reduce 3SAT ϕ to MAX2SAT.

MAX2SAT Is NP-Complete[a]

- Consider the following 10 clauses:

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

- Let the 2SAT formula $r(x, y, z, w)$ represent the conjunction of these clauses.
- The clauses are symmetric with respect to x, y, and z.
- How many clauses can we satisfy?

[a] Garey, Johnson, and Stockmeyer (1976).

The Proof (continued)

All of x, y, z are true: By setting w to true, we satisfy $4 + 0 + 3 = 7$ clauses, whereas by setting w to false, we satisfy only $3 + 0 + 3 = 6$ clauses.

Two of x, y, z are true: By setting w to true, we satisfy $3 + 2 + 2 = 7$ clauses, whereas by setting w to false, we satisfy $2 + 2 + 3 = 7$ clauses.

The Proof (continued)

One of x, y, z is true: By setting w to false, we satisfy $1 + 3 + 3 = 7$ clauses, whereas by setting w to true, we satisfy only $2 + 3 + 1 = 6$ clauses.

None of x, y, z is true: By setting w to false, we satisfy $0 + 3 + 3 = 6$ clauses, whereas by setting w to true, we satisfy only $1 + 3 + 0 = 4$ clauses.

The Proof (continued)

- A truth assignment that satisfies $x \vee y \vee z$ can be *extended* to satisfy 7 of the 10 clauses of $r(x, y, z, w)$, *and no more.*
- A truth assignment that does not satisfy $x \vee y \vee z$ can be extended to satisfy only 6 of them, *and no more.*
- The reduction from 3SAT ϕ to MAX2SAT $R(\phi)$:
  - For each clause $C_i = (\alpha \vee \beta \vee \gamma)$ of ϕ, add **group** $r(\alpha, \beta, \gamma, w_i)$ to $R(\phi)$.
- If ϕ has m clauses, then $R(\phi)$ has $10m$ clauses.
- Finally, set $K = 7m$.

The Proof (concluded)

- We now show that K clauses of $R(\phi)$ can be satisfied if and only if ϕ is satisfiable.
- Suppose $7m$ clauses of $R(\phi)$ can be satisfied.
  - 7 clauses must be satisfied in each group because each group can have at most 7 clauses satisfied.
  - Hence all clauses of ϕ must be satisfied.[a]
- Suppose all clauses of ϕ are satisfied.
  - Each group can set its w_i appropriately to have 7 clauses satisfied.

[a] If 70% of the world population are male and if at most 70% of each country's population are male, then each country must have exactly 70% male population.

Michael R. Garey (1945–)

David S. Johnson (1945–)

Larry Stockmeyer (1948–2004)

NAESAT

- The NAESAT (for "not-all-equal" SAT) is like 3SAT.
- But there must be a satisfying truth assignment under which no clauses have the three literals equal in truth value.
- Equivalently, there is a truth assignment such that each clause has one literal assigned true and one literal assigned false.

NAESAT Is NP-Complete[a]

- Recall the reduction of CIRCUIT SAT to SAT on p. 231ff.
- It produced a CNF ϕ in which each clause has 1, 2, or 3 literals.
- Add the same variable z to all clauses with fewer than 3 literals to make it a 3SAT formula.
- Goal: The new formula $\phi(z)$ is NAE-satisfiable if and only if the original circuit is satisfiable.

[a]Karp (1972).

The Proof (continued)

- Suppose T NAE-satisfies $\phi(z)$.
  - $\bar{T}$ also NAE-satisfies $\phi(z)$.
  - Under T or $\bar{T}$, variable z takes the value false.
  - This truth assignment $\mathcal{T}$ must satisfy all the clauses of ϕ.
    - Because z is not the reason that makes $\phi(z)$ true under $\mathcal{T}$.
  - So $\mathcal{T} \models \phi$.
  - So the original circuit is satisfiable.

The Proof (concluded)

- Suppose there is a truth assignment that satisfies the circuit.
  - Then there is a truth assignment T that satisfies every clause of ϕ.
  - Extend T by adding $T(z) = \texttt{false}$ to obtain T'.
  - T' satisfies $\phi(z)$.
  - So in no clauses are all three literals false under T'.
  - In no clauses are all three literals true under T'.
    - Need to review the detailed construction on p. 232 and p. 233.

Richard Karp[a] (1935–)

[a]Turing Award (1985).

Undirected Graphs

- An **undirected graph** $G = (V, E)$ has a finite set of nodes, V, and a set of *undirected* edges, E.
- It is like a directed graph except that the edges have no directions and there are no self-loops.
- Use $[\, i, j \,]$ to denote the fact that there is an edge between node i and node j.

Independent Sets

- Let $G = (V, E)$ be an undirected graph.
- $I \subseteq V$.
- I is **independent** if there is no edge between any two nodes $i, j \in I$.
- The INDEPENDENT SET problem: Given an undirected graph and a goal K, is there an independent set of size K?
- Many applications.

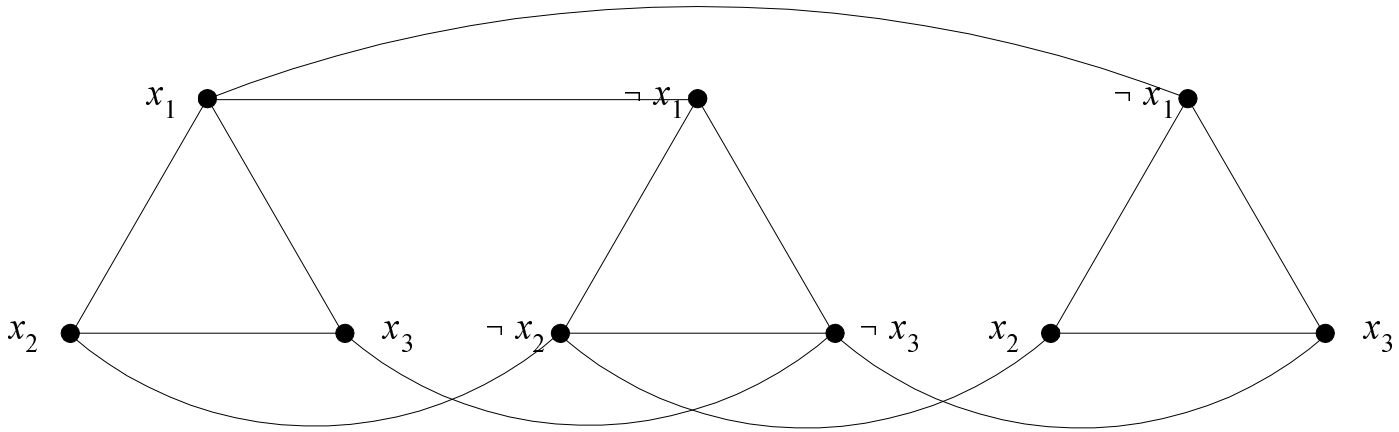
INDEPENDENT SET Is NP-Complete

- This problem is in NP: Guess a set of nodes and verify that it is independent and meets the count.
- We will reduce 3SAT to INDEPENDENT SET.
- If a graph contains a triangle, any independent set can contain at most one node of the triangle.
- The results of the reduction will be graphs whose nodes can be partitioned into m disjoint triangles.
  - If the special case of graphs is hard, the original problem must be at least as hard (why?).

The Proof (continued)

- Let ϕ be an instance of 3SAT with m clauses.
- We will construct graph G (with constraints as said) with $K = m$.
- Furthermore, ϕ is satisfiable if and only if G has an independent set of size K.
- There is a triangle for each clause with the literals as the nodes.
- Then add edges between x and $\neg x$ for every variable x.

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



Same literals that appear in different clauses yield *distinct* nodes.

The Proof (continued)

- Suppose G has an independent set I of size $K = m$.
  - An independent set can contain at most m nodes, one from each triangle.
  - So I contains exactly one node from each triangle.
  - Truth assignment T assigns true to those literals in I.
  - T is consistent because contradictory literals are connected by an edge; hence both cannot be in I.
  - T satisfies ϕ because it has a node from every triangle, thus satisfying every clause.[a]

[a] The variables without a truth value can be assigned arbitrarily. Contributed by Mr. Chun-Yuan Chen (`R99922119`) on November 2, 2010.

The Proof (concluded)

- Suppose a satisfying truth assignment T exists for ϕ.
  - Collect one node from each triangle whose literal is true under T.
  - The choice is arbitrary if there is more than one true literal.
  - This set of m nodes must be independent by construction.
    - Both literals x and $\neg x$ cannot be assigned true.

Other INDEPENDENT SET-Related NP-Complete Problems

Corollary 37 INDEPENDENT SET *is NP-complete for 4-degree graphs.*

Theorem 38 INDEPENDENT SET *is NP-complete for planar graphs.*

Theorem 39 (Garey and Johnson (1977)) INDEPENDENT SET *is NP-complete for 3-degree planar graphs.*

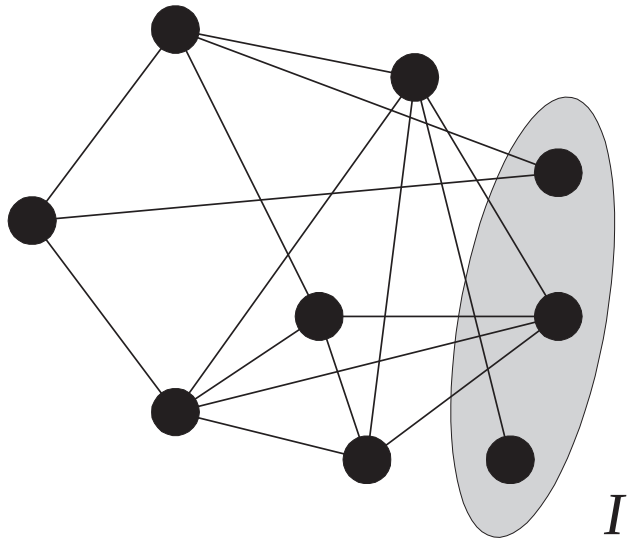
NODE COVER

- We are given an undirected graph G and a goal K.
- NODE COVER: Is there a set C with K or fewer nodes such that each edge of G has at least one of its endpoints in C?
- Many applications.

NODE COVER Is NP-Complete

Corollary 40 NODE COVER *is NP-complete.*

- I is an independent set of $G = (V, E)$ if and only if $V - I$ is a node cover of G.



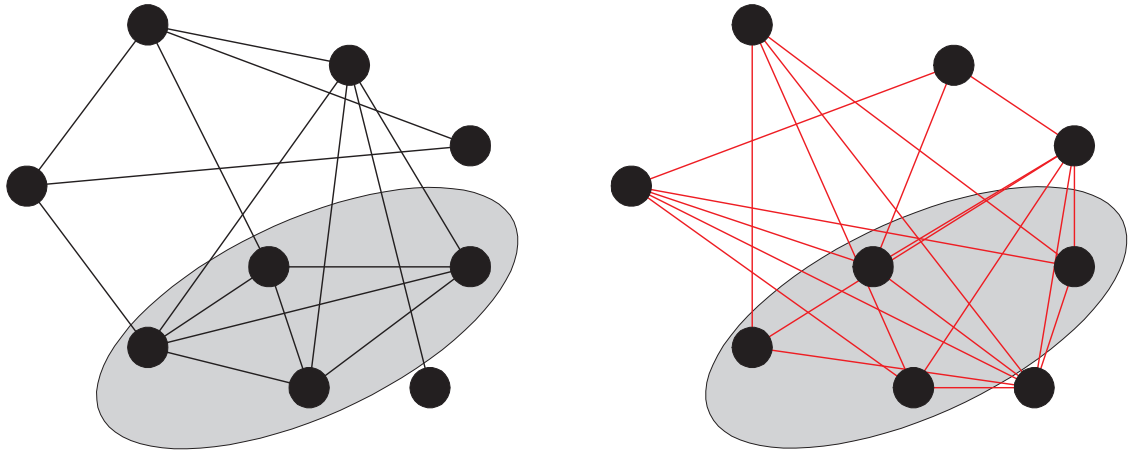
CLIQUE

- We are given an undirected graph G and a goal K.
- CLIQUE asks if there is a set C with K nodes such that there is an edge between any two nodes $i, j \in C$.
- Many applications.

CLIQUE Is NP-Complete

Corollary 41 CLIQUE *is NP-complete.*

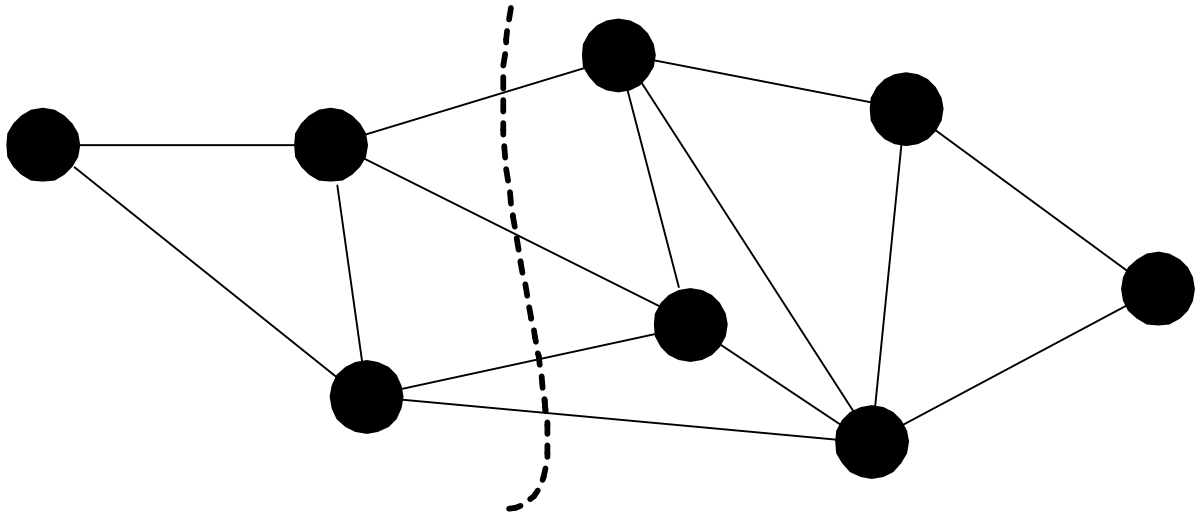
- Let $\bar{G}$ be the **complement** of G, where $[x, y] \in \bar{G}$ if and only if $[x, y] \notin G$.
- I is a clique in $G \Leftrightarrow I$ is an independent set in $\bar{G}$.



MIN CUT and MAX CUT

- A **cut** in an undirected graph $G = (V, E)$ is a partition of the nodes into two nonempty sets S and $V - S$.
- The size of a cut $(S, V - S)$ is the number of edges between S and $V - S$.
- MIN CUT $\in$ P by the maxflow algorithm.
- MAX CUT asks if there is a cut of size at least K.
  - K is part of the input.

A Cut of Size 4



MIN CUT and MAX CUT (concluded)

- MAX CUT has applications in circuit layout.
  - The minimum area of a VLSI layout of a graph is not less than the square of its maximum cut size.[a]

[a] Raspaud, Sýkora, and Vrťo (1995); Mak and Wong (2000).

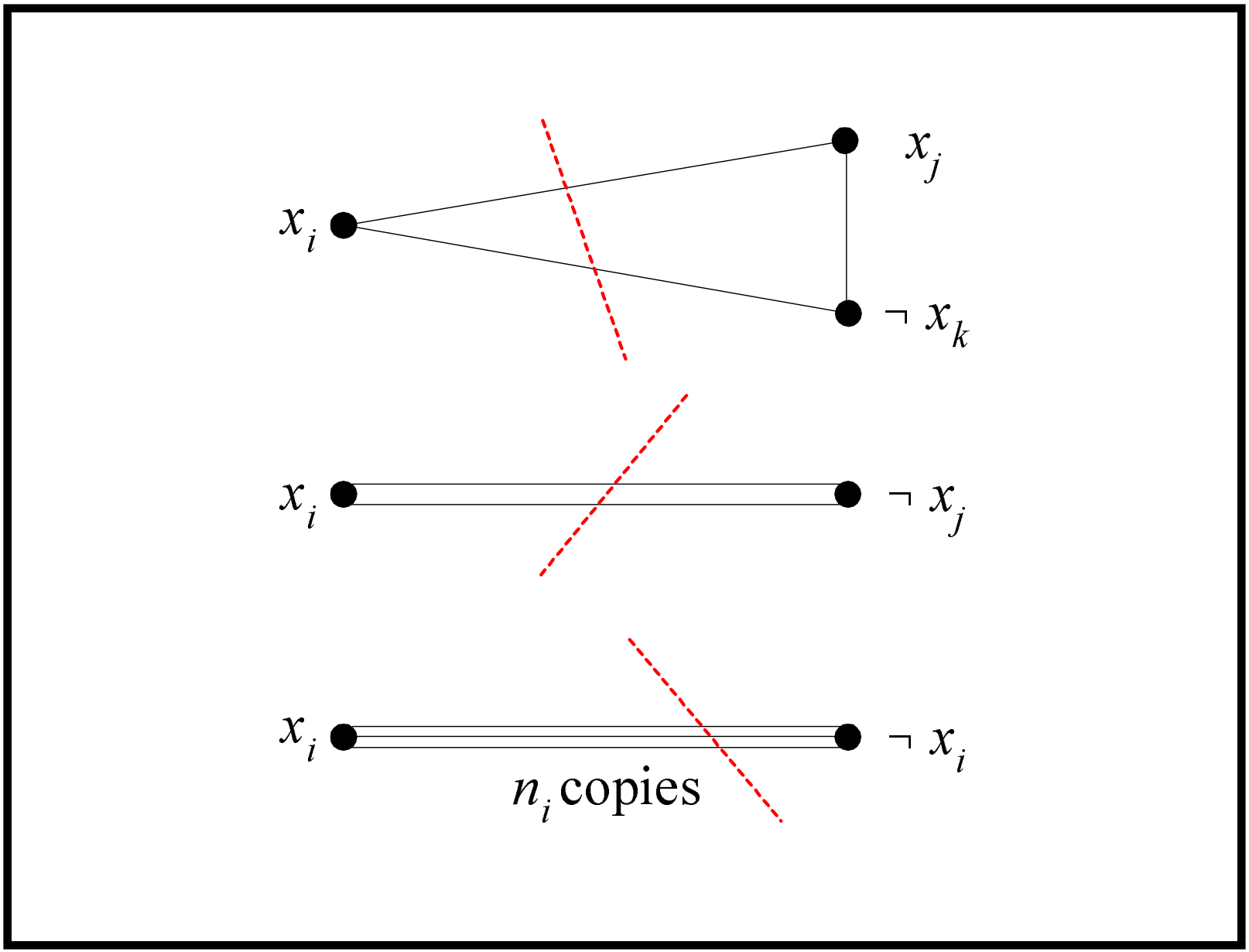
MAX CUT Is NP-Complete[a]

- We will reduce NAESAT to MAX CUT.
- Given an instance ϕ of 3SAT with m clauses, we shall construct a graph $G = (V, E)$ and a goal K.
- Furthermore, there is a cut of size at least K if and only if ϕ is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes one to the cut if its nodes are separated.

[a]Garey, Johnson, and Stockmeyer (1976).

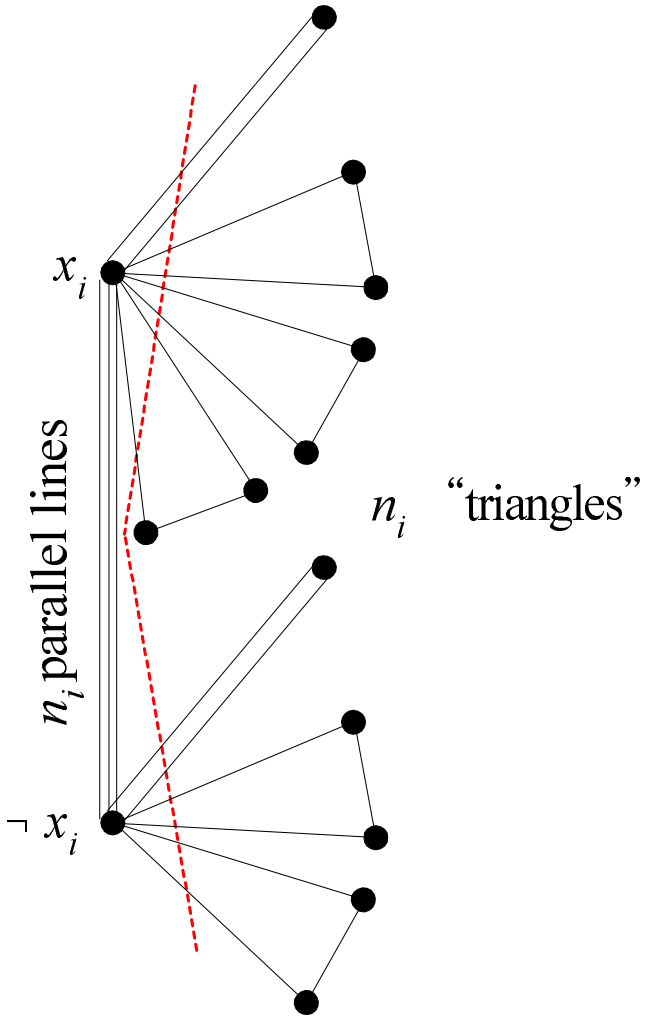
The Proof

- Suppose ϕ's m clauses are $C_1, C_2, \ldots, C_m$.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- G has $2n$ nodes: $x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n$.
- Each clause with 3 distinct literals makes a triangle in G.
- For each clause with two identical literals, there are two parallel edges between the two distinct literals.
- No need to consider clauses with one literal (why?).
- For each variable x_i, add n_i copies of edge $[x_i, \neg x_i]$, where n_i is the number of occurrences of x_i and $\neg x_i$ in ϕ.


x_i
x_j
$\neg x_k$
x_i
$\neg x_j$
x_i
$\neg x_i$
n_i copies

The Proof (continued)

- Set $K = 5m$.
- Suppose there is a cut $(S, V - S)$ of size $5m$ or more.
- A clause (a triangle or two parallel edges) contributes at most 2 to a cut no matter how you split it.
- Suppose both x_i and $\neg x_i$ are on the same side of the cut.
- They *together* contribute at most $2n_i$ edges to the cut.
  - They appear in at most n_i different clauses.
  - A clause contributes at most 2 to a cut.



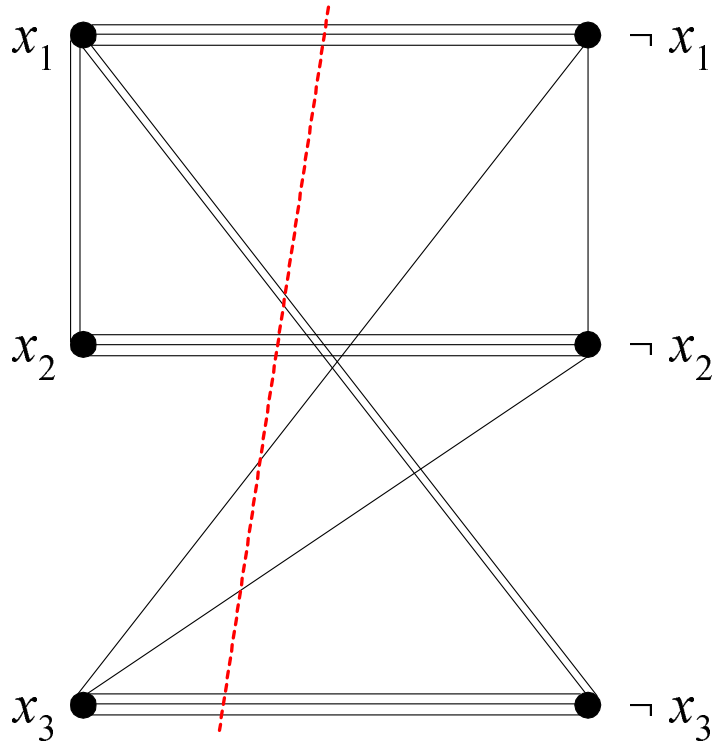
The Proof (continued)

- Either x_i or $\neg x_i$ contributes at most n_i to the cut by the pigeonhole principle.
- Changing the side of that literal does not decrease the size of the cut.
- Hence we assume variables are separated from their negations.
- The total number of edges in the cut that join opposite literals x_i and $\neg x_i$ is $\sum_{i=1}^{n} n_i$.
- But $\sum_{i=1}^{n} n_i = 3m$ as it is simply the total number of literals.

The Proof (concluded)

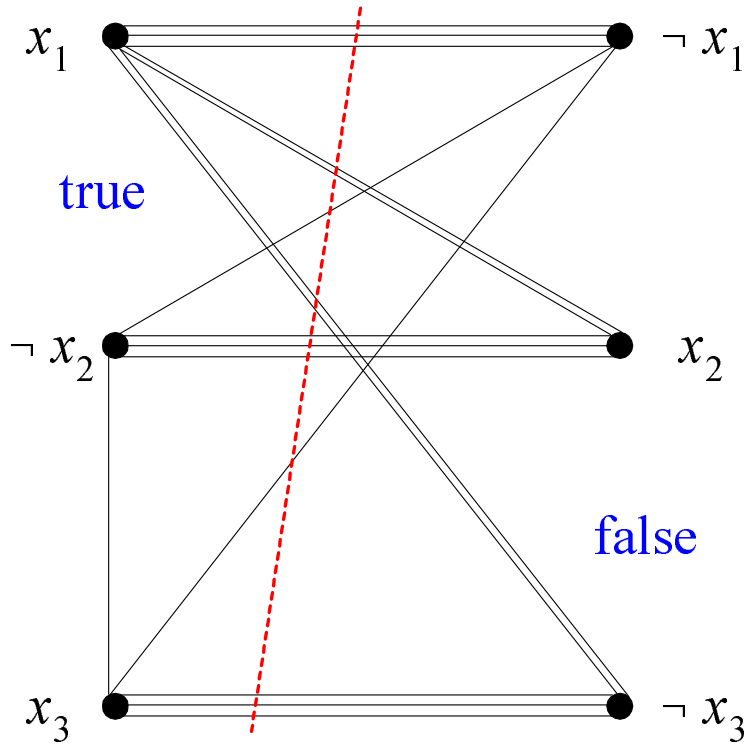
- The *remaining* $K - 3m = 2m$ edges in the cut must come from the m triangles or parallel edges that correspond to the clauses.
- Each can contribute at most 2 to the cut.
- So all are split.
- A split clause means at least one of its literals is true and at least one false.
- The other direction is left as an exercise.

A Cut That Does Not Meet the Goal $K = 5 \times 3 = 15$



- $(x_1 \vee x_2 \vee x_2) \wedge (x_1 \vee \neg x_3 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$.
- The cut size is $13 < 15$.

A Cut That Meets the Goal $K = 5 \times 3 = 15$



- $(x_1 \vee x_2 \vee x_2) \wedge (x_1 \vee \neg x_3 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$.
- The cut size is now 15.

Remarks

- We had proved that MAX CUT is NP-complete for multigraphs.
- How about proving the same thing for simple graphs?[a]
- How to modify the proof to reduce 4SAT to MAX CUT?[b]
- All NP-complete problems are mutually reducible by definition as an NP-complete problem is in NP.[c]
  - So they are equally hard in this sense.[d]

---

[a] Contributed by Mr. Tai-Dai Chou (J93922005) on June 2, 2005.
[b] Contributed by Mr. Chien-Lin Chen (J94922015) on June 8, 2006.
[c] Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.
[d] Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.