

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

- From Propositions 27 (p. 272) and Proposition 30 (p. 275), 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 (1998)

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

## Use of Reduction in Proving NP-Completeness

- Recall that  $L_1$  reduces to  $L_2$  if there is an efficient function  $R$  such that for all inputs  $x$  (p. 247),

$$x \in L_1 \text{ if and only if } R(x) \in L_2.$$

- When  $L_1$  is known to be NP-complete and when  $L_2 \in \text{NP}$ , then  $L_2$  is NP-complete.<sup>a</sup>
- A common mistake is to focus on solving  $x \in L_1$  or solving  $R(x) \in L_2$ .
- The correct way is to, given a certificate for  $x \in L_1$  (a satisfying truth assignment, e.g.), construct a certificate for  $R(x) \in L_2$  (a Hamiltonian path, e.g.), and vice versa.

---

<sup>a</sup>Because NP is closed under reductions (p. 271).

## 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. 295) and the reduction of CIRCUIT SAT to SAT (p. 261).
- 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.
  - So

$$x_1 \vee x_2 \quad \text{becomes} \quad x_1 \vee x_2 \vee x_2.$$

## The Satisfiability of Random 3SAT Expressions

- Consider a random 3SAT expressions  $\phi$  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  $\phi$  is satisfiable as more constraints are added.
- Indeed, there is a  $c_n$  such that for  $c < c_n(1 - \epsilon)$ ,  $\phi$  is satisfiable almost surely, and for  $c > c_n(1 + \epsilon)$ ,  $\phi$  is unsatisfiable almost surely.<sup>a</sup>

---

<sup>a</sup>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.)*

## The Proof (continued)

- 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.
  - $x_1, x_2, \dots, x_k$  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  $\neg x_i \vee x_j$  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  $\phi$  be an instance of 2SAT: Each clause has 2 literals.
- NL is a subset of P (p. 229).
- By Eq. (2) on p. 239, coNL equals NL.
- We need to show only that recognizing unsatisfiable expressions is in NL.
- See the textbook for the complete proof.

## Generalized 2SAT: MAX2SAT

- Consider a 2SAT expression.
- Let  $K \in \mathbb{N}$ .
- MAX2SAT asks whether there is a truth assignment that satisfies at least  $K$  of the clauses.
  - MAX2SAT becomes 2SAT when  $K$  equals the number of clauses.

## Generalized 2SAT: MAX2SAT (concluded)

- MAX2SAT is an optimization problem.
  - With binary search, one can nail the maximum number of satisfiable clauses of the 2SAT expression.
- MAX2SAT  $\in$  NP: Guess a truth assignment and verify the count.
- We now reduce 3SAT  $\phi$  to MAX2SAT.

## MAX2SAT Is NP-Complete<sup>a</sup>

- 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?

---

<sup>a</sup>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  $\phi$  to MAX2SAT  $R(\phi)$ :
  - For each clause  $C_i = (\alpha \vee \beta \vee \gamma)$  of  $\phi$ , add **group**  $r(\alpha, \beta, \gamma, w_i)$  to  $R(\phi)$ .
- If  $\phi$  has  $m$  clauses, then  $R(\phi)$  has  $10m$  clauses.
- Finally, set  $K = 7m$ .

## The Proof (continued)

- We now show that  $K$  clauses of  $R(\phi)$  can be satisfied if and only if  $\phi$  is satisfiable.
- Suppose  $K = 7m$  clauses of  $R(\phi)$  can be satisfied.
  - 7 clauses of each group  $r(\alpha, \beta, \gamma, w_i)$  must be satisfied because each group can have at most 7 clauses satisfied.<sup>a</sup>
  - Hence each clause  $C_i = (\alpha \vee \beta \vee \gamma)$  of  $\phi$  is satisfied by the same truth assignment.
  - So  $\phi$  is satisfied.

---

<sup>a</sup>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.

## The Proof (concluded)

- Suppose  $\phi$  is satisfiable.
  - Let  $T$  satisfy all clauses of  $\phi$ .
  - Each group  $r(\alpha, \beta, \gamma, w_i)$  can set its  $w_i$  appropriately to have 7 clauses satisfied.
  - So  $K = 7m$  clauses are satisfied.

## 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 all 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<sup>a</sup>

- Recall the reduction of CIRCUIT SAT to SAT on p. 261ff.
- It produced a CNF  $\phi$  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.

---

<sup>a</sup>Karp (1972).

## The Proof (continued)

- Suppose  $T$  NAE-satisfies  $\phi(z)$ .
  - $\bar{T}$  takes the opposite truth value of  $T$  on every variable.
  - $\bar{T}$  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  $\phi$ .
    - \* 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  $\phi$ .
  - Extend  $T$  by adding  $T(z) = \mathbf{false}$  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. 262 and p. 263.

## Richard Karp<sup>a</sup> (1935–)



---

<sup>a</sup>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.

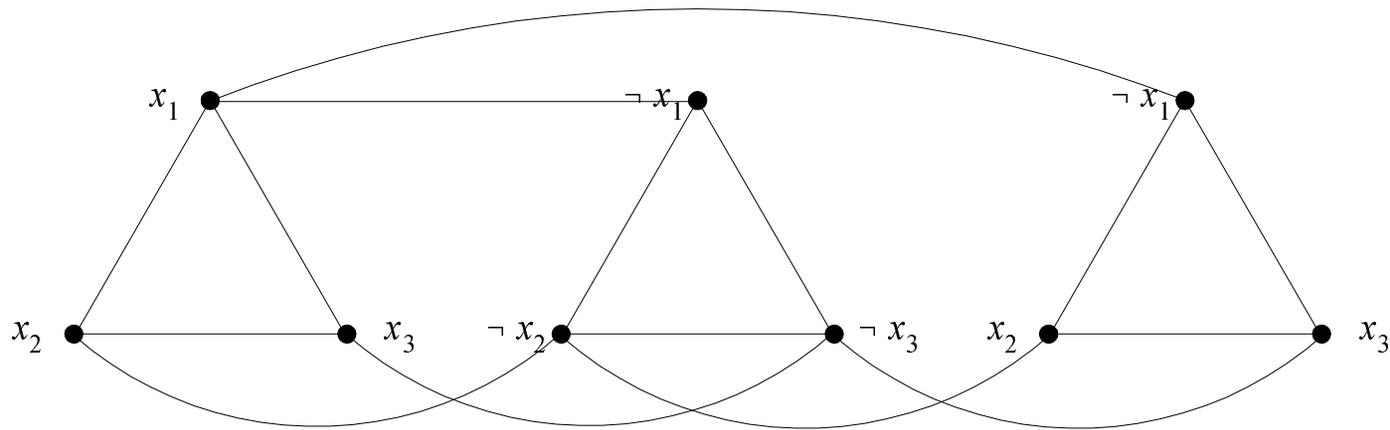
## INDEPENDENT SET Is NP-Complete

- 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.

## The Proof (continued)

- Let  $\phi$  be an instance of 3SAT with  $m$  clauses.
- We will construct graph  $G$  with  $K = m$ .
- Furthermore,  $\phi$  is satisfiable if and only if  $G$  has an independent set of size  $K$ .
- Here is the reduction:
  - There is a triangle for each clause with the literals as the nodes.
  - 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 literal labels 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  $\phi$  because it has a node from every triangle, thus satisfying every clause.<sup>a</sup>

---

<sup>a</sup>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  $\phi$  is a satisfiable.
  - Let truth assignment  $T$  satisfy  $\phi$ .
  - 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.*

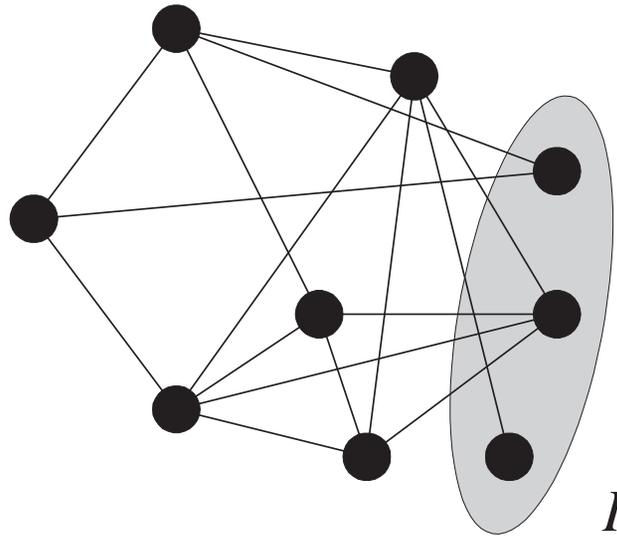
## NODE COVER

- 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 (i.e., incident nodes) in  $C$ ?
- Many applications.

## NODE COVER Is NP-Complete

**Corollary 40 (Karp (1972))** 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  $G$ .



## Remarks<sup>a</sup>

- Are INDEPENDENT SET and NODE COVER NP-complete if  $K$  is a constant?
  - No, because one can do an exhaustive search on all the possible node covers or independent sets (both  $\binom{n}{K}$  of them, a polynomial).<sup>b</sup>
- Are INDEPENDENT SET and NODE COVER NP-complete if  $K$  is a linear function of  $n$ ?
  - INDEPENDENT SET with  $K = n/3$  and NODE COVER with  $K = 2n/3$  remain NP-complete by our reductions.

---

<sup>a</sup>Contributed by Mr. Ching-Hua Yu (D00921025) on October 30, 2012.

<sup>b</sup> $n = |V|$ .

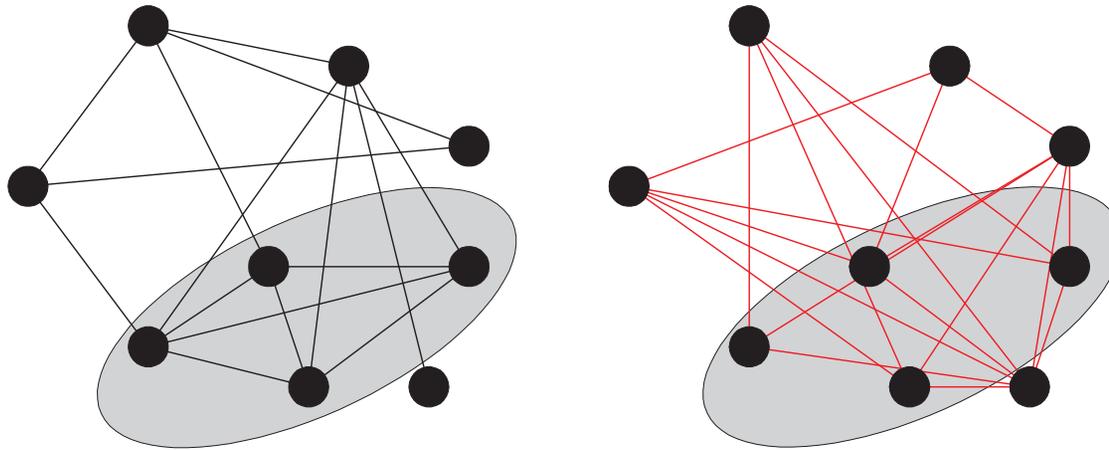
## CLIQUE

- 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 (Karp (1972))** *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  $\bar{G}$ .



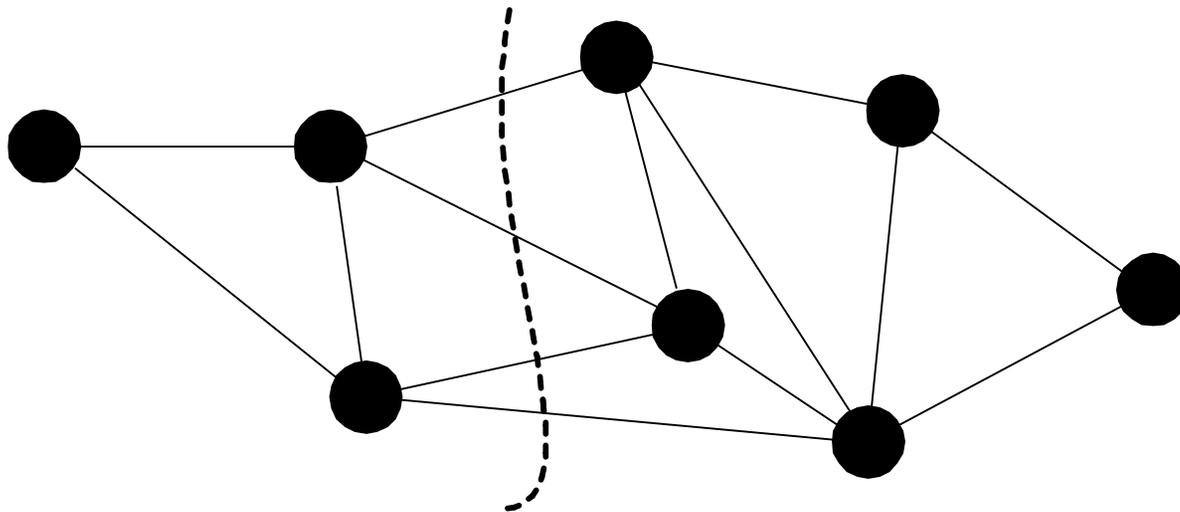
## MIN CUT and MAX CUT

- 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.<sup>a</sup>
- MAX CUT asks if there is a cut of size at least  $K$ .
  - $K$  is part of the input.

---

<sup>a</sup>In time  $O(|V| \cdot |E|)$  by Orlin (2012).

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.<sup>a</sup>

---

<sup>a</sup>Raspaud, Sýkora, and Vrřo (1995); Mak and Wong (2000).

## MAX CUT Is NP-Complete<sup>a</sup>

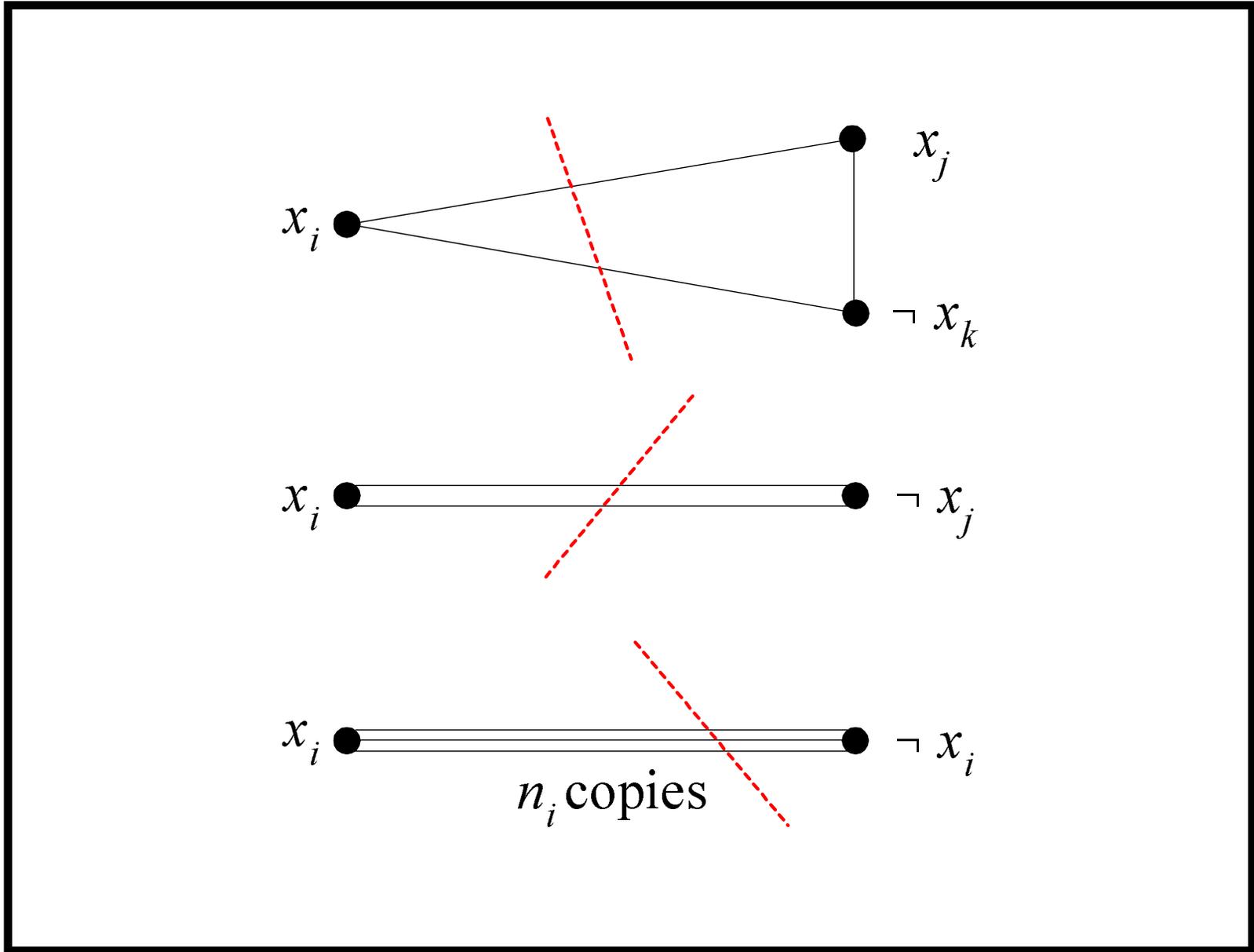
- We will reduce NAESAT to MAX CUT.
- Given an instance  $\phi$  of 3SAT with  $m$  clauses, we shall construct a graph  $G = (V, E)$  and a goal  $K$ .
- Furthermore, there is a cut of size at least  $K$  if and only if  $\phi$  is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.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

---

<sup>a</sup>Karp (1972) and Garey, Johnson, and Stockmeyer (1976).

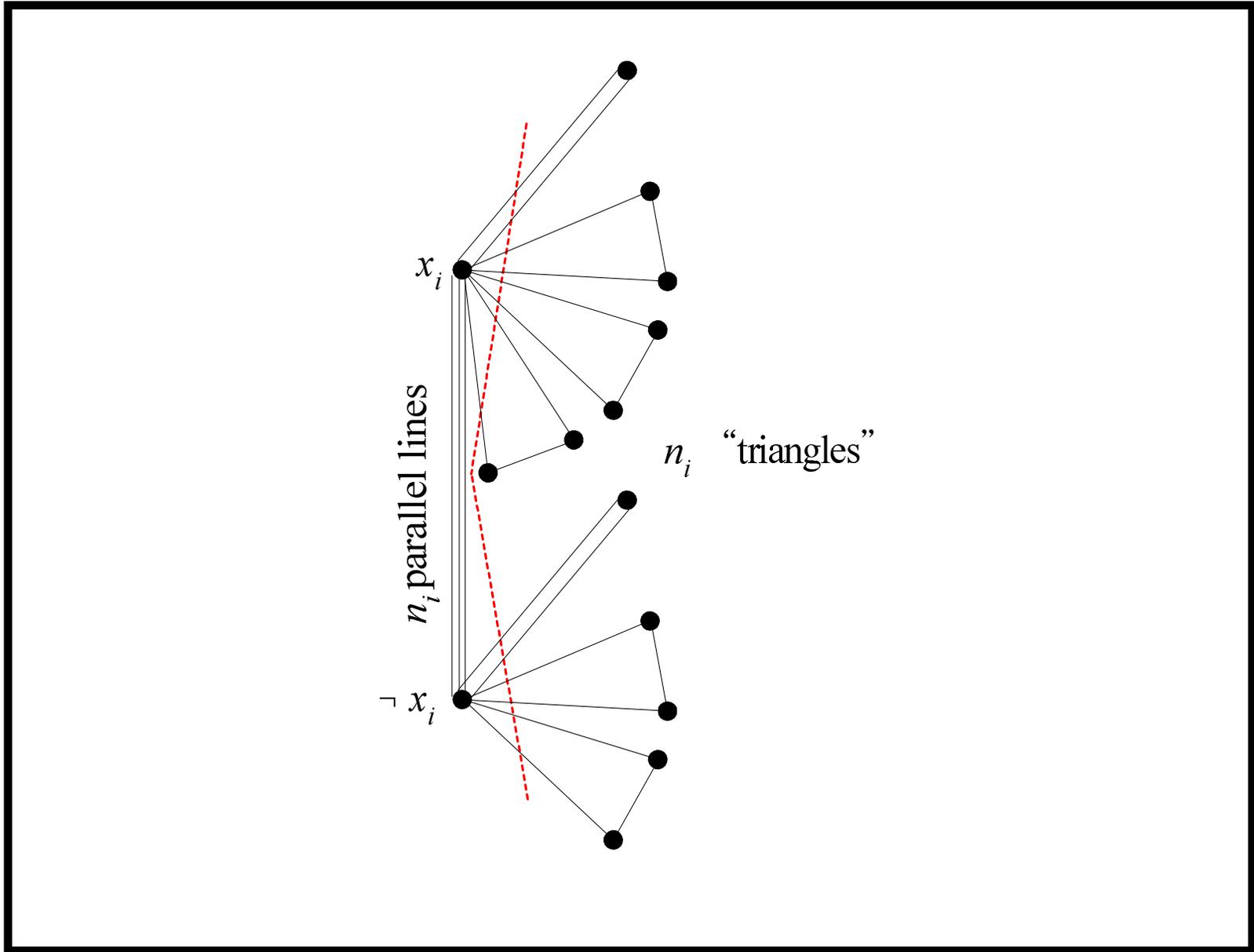
## The Proof

- Suppose  $\phi$ 's  $m$  clauses are  $C_1, C_2, \dots, C_m$ .
- The boolean variables are  $x_1, x_2, \dots, x_n$ .
- $G$  has  $2n$  nodes:  $x_1, x_2, \dots, x_n, \neg x_1, \neg x_2, \dots, \neg x_n$ .
- 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  $\phi$ .



## 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.

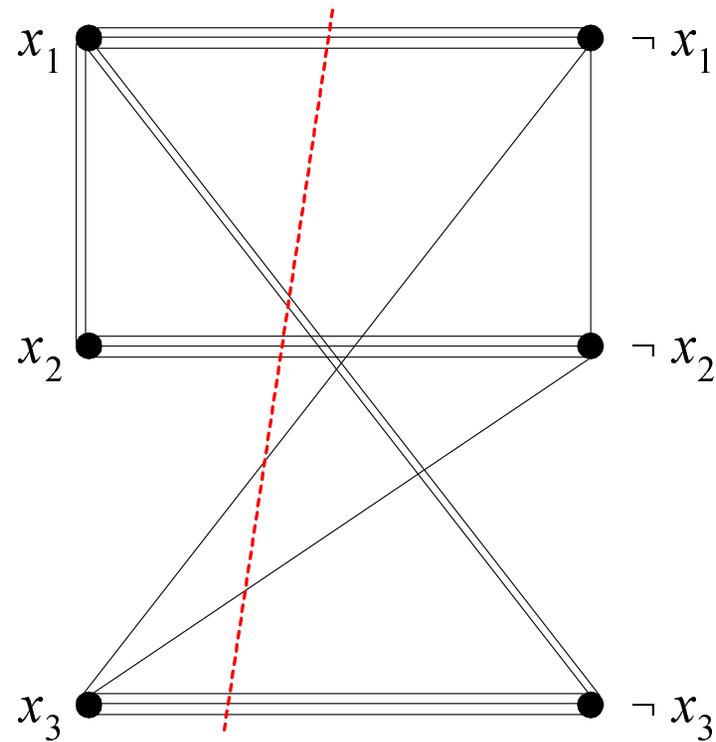
## The Proof (concluded)

- The *remaining*  $K - 3m \geq 2m$  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.<sup>a</sup>
- 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.

---

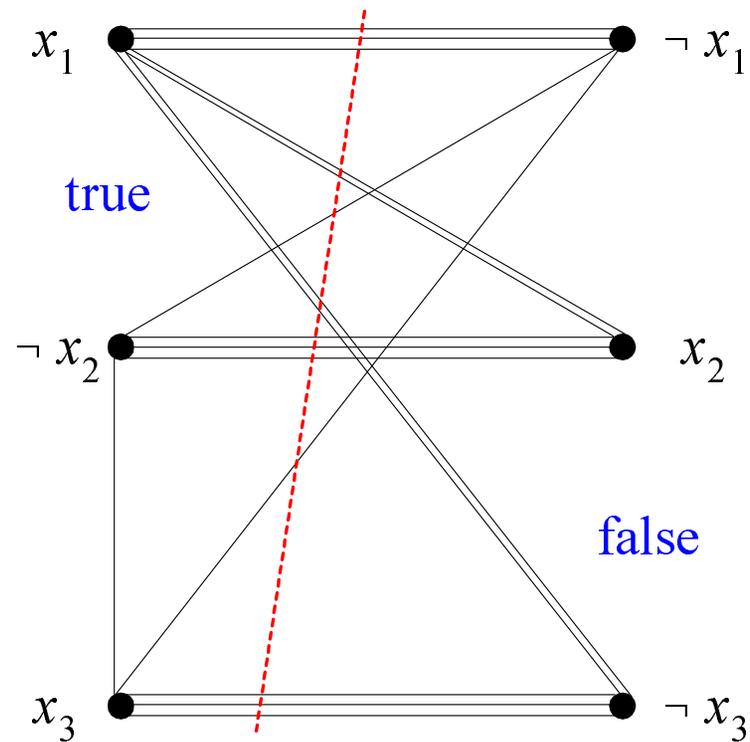
<sup>a</sup>So  $K = 5m$ .

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?<sup>a</sup>
- How to modify the proof to reduce 4SAT to MAX CUT?<sup>b</sup>
- All NP-complete problems are mutually reducible by definition.<sup>c</sup>
  - So they are equally hard in this sense.<sup>d</sup>

---

<sup>a</sup>Contributed by Mr. Tai-Dai Chou (J93922005) on June 2, 2005.

<sup>b</sup>Contributed by Mr. Chien-Lin Chen (J94922015) on June 8, 2006.

<sup>c</sup>Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.

<sup>d</sup>Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.