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.

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.
- How many clauses can we satisfy?
- The clauses are symmetric with respect to x, y, and z.

[a] Garey, Johnson, Stockmeyer, 1976.

The Proof (continued)

All of x, y, z are true: By setting w to true, we *can* satisfy $4 + 0 + 3 = 7$ clauses.

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

One of x, y, z is true: By setting w to false, we *can* satisfy $1 + 3 + 3 = 7$ clauses.

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

©2002 Yuh-Dauh Lyuu, National Taiwan University

The Proof (continued)

- Any truth assignment that satisfies $x \vee y \vee z$ can be extended to satisfy 7 of the 10 clauses and no more.
- Any other truth assignment can be extended to satisfy only 6 of them.
- 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.
- 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.
- Suppose all clauses of ϕ are satisfied.
  - Each group can set its w_i appropriately to have 7 clauses satisfied.

NAESAT

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

NAESAT Is NP-Complete[a]

- Recall the reduction of CIRCUIT SAT to SAT on p. 203.
- It produced a CNF ϕ in which each clause has at most 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 must still satisfy all clauses of ϕ.
  - So it satisfies the original circuit.

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'.
  - Under T', in no clauses are all three literals true.
    * Review the construction on p. 204 and p. 205.

©2002 Yuh-Dauh Lyuu, National Taiwan University

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.
- We 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 whenever $i, j \in I$, there is no edge between i and j.
- The INDEPENDENT SET problem: Given an undirected graph and a goal K, is there an independent set of size K?
  - Many applications.

©2002 Yuh-Dauh Lyuu, National Taiwan University

INDEPENDENT SET Is NP-Complete

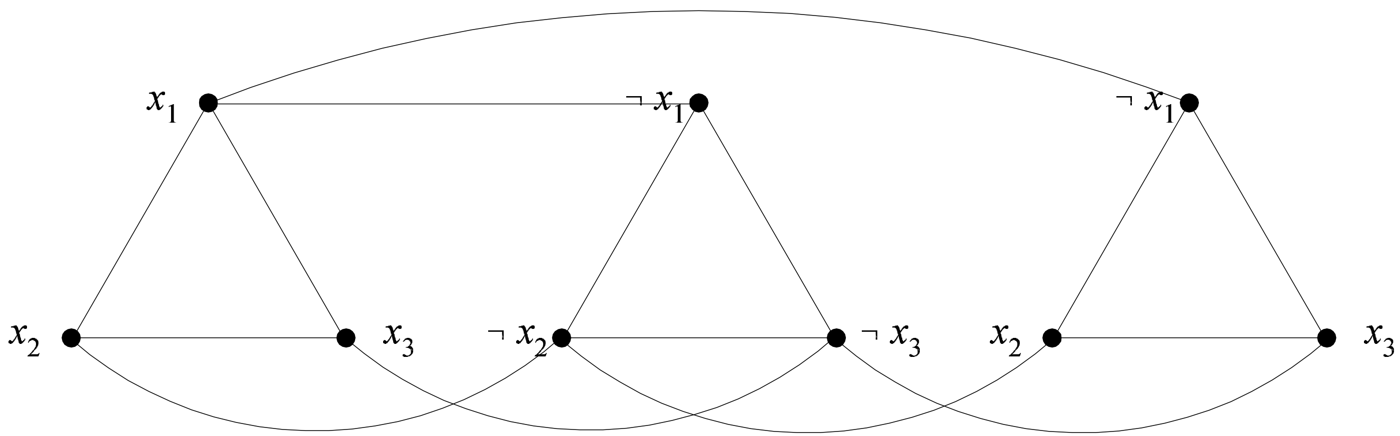
- This problem is in NP: Guess a set of nodes and verify that it is independent and meets the count.
- If a graph contains a triangle, any independent set can contain at most one node of the triangle.
- We consider graphs whose nodes can be partitioned in m disjoint triangles.
  - If the special case is hard, the original problem must be at least as hard.

©2002 Yuh-Dauh Lyuu, National Taiwan University

Reduction from 3SAT to INDEPENDENT SET

- Let ϕ be an instance of 3SAT with m clauses.
- We will construct graph G (with constraints as said) with $K = m$ such that 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.
- Add additional edges between x and $\neg x$ for every variable x.

A Sample Construction



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

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.
  - An independent set of size m exists if and only if it 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 not both in I.
  - T satisfies ϕ because it has a node from every triangle, thus satisfying every clause.

The Proof (concluded)

- Suppose a satisfying truth assignment T exists for ϕ.
  - Collect one node from each triangle whose literal is true under T.
  - This set of m nodes must be independent by construction.

Corollary 36 4-DEGREE INDEPENDENT SET *is NP-complete.*

Theorem 37 INDEPENDENT SET *is NP-complete for planar graphs.*

CLIQUE and NODE COVER

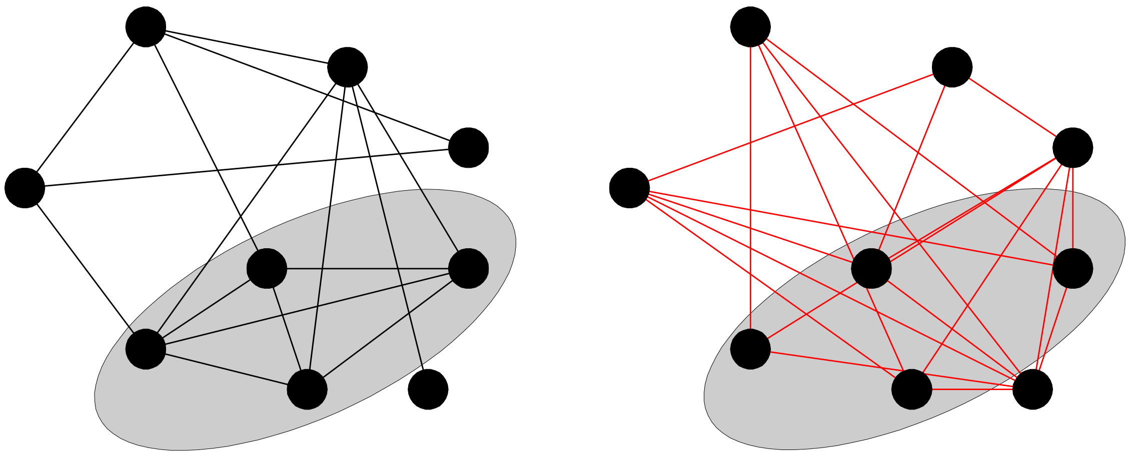
- We are given an undirected graph G and a goal K.
- CLIQUE asks if there is a set of K nodes that form a **clique**, which have all possible edges between them.
- NODE COVER asks if there is a set C with K or fewer nodes such that each edge of G has at least one of its endpoints in C.

©2002 Yuh-Dauh Lyuu, National Taiwan University

CLIQUE Is NP-Complete

Corollary 38 CLIQUE *is NP-complete.*

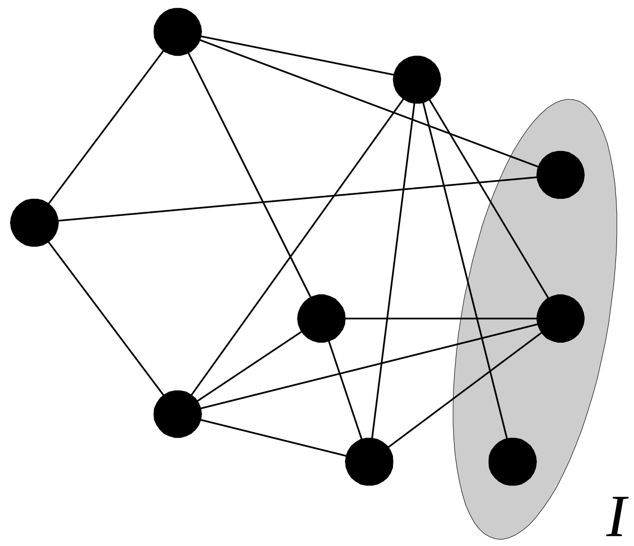
- Let $\bar{G}$ be the **complement** of G, where $[x, y] \in \bar{G}$ if and only if $[x, y] \notin G$.
- I is a clique in $G \Leftrightarrow I$ is an independent set in $\bar{G}$.



NODE COVER Is NP-Complete

Corollary 39 NODE COVER *is NP-complete.*

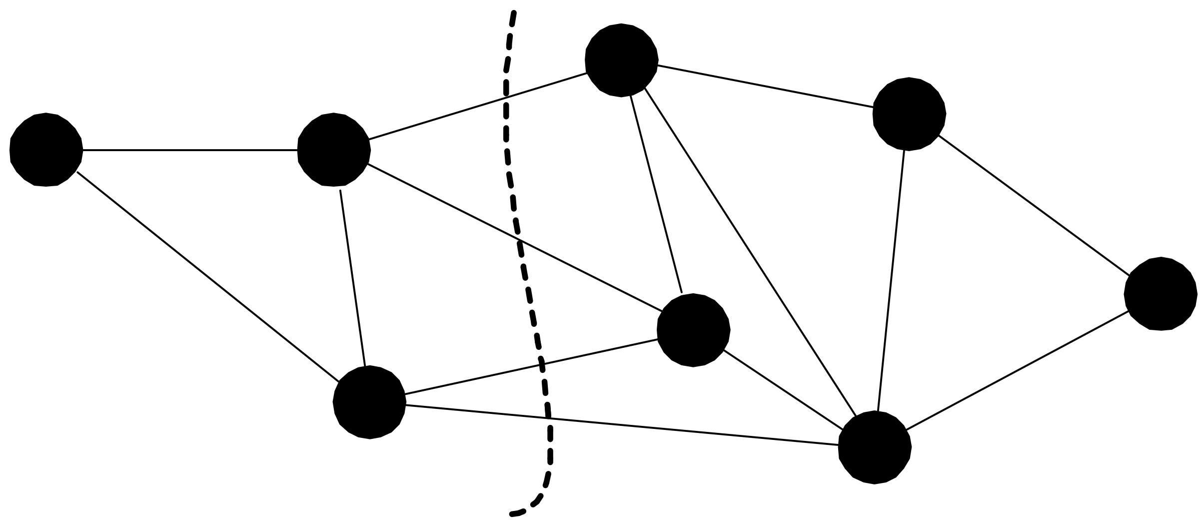
- I is an independent set of $G = (V, E)$ if and only if $V - I$ is a node cover of G.



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



MAX CUT Is NP-Complete[a]

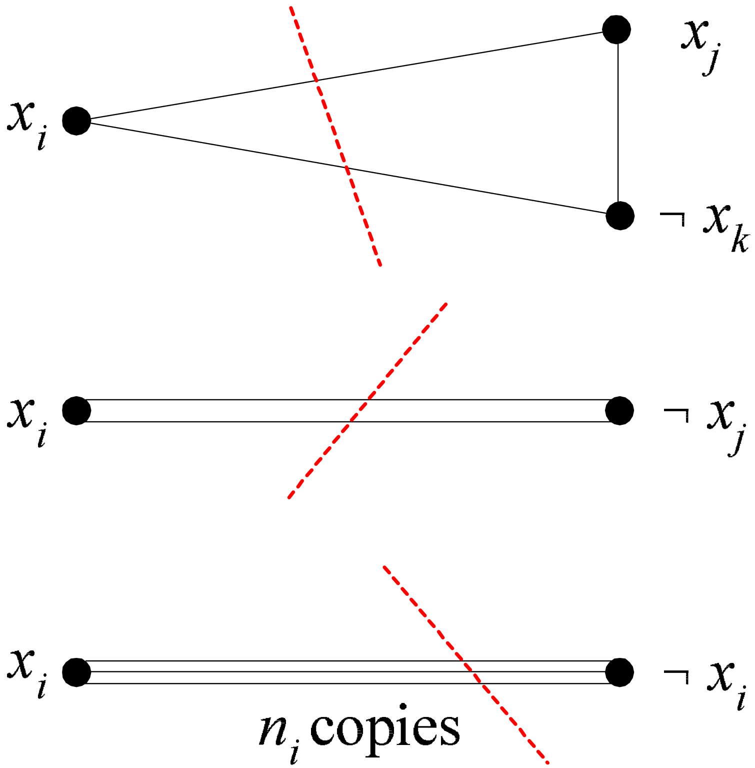
- We will reduce NAESAT to MAX CUT.
- Given an instance ϕ of 3SAT with m clauses, we shall construct a graph $G = (V, E)$ and a goal K such that:
  - There is a cut of size at least K if and only if ϕ is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes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

[a]Garey, Johnson, Stockmeyer, 1976.

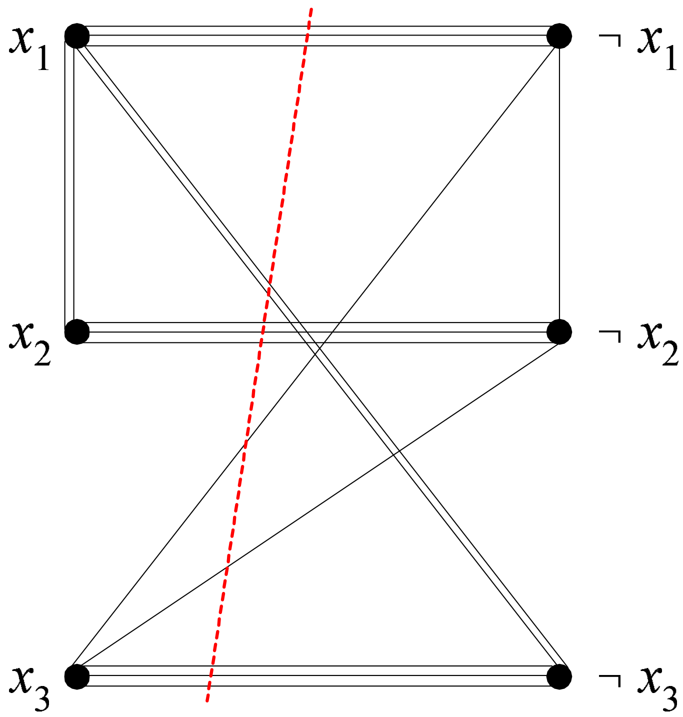
Reduction from NAESAT to MAX CUT

- 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 the edge $[x_i, \neg x_i]$, where n_i is the number of occurrences of x_i and $\neg x_i$ in ϕ.

The Construction



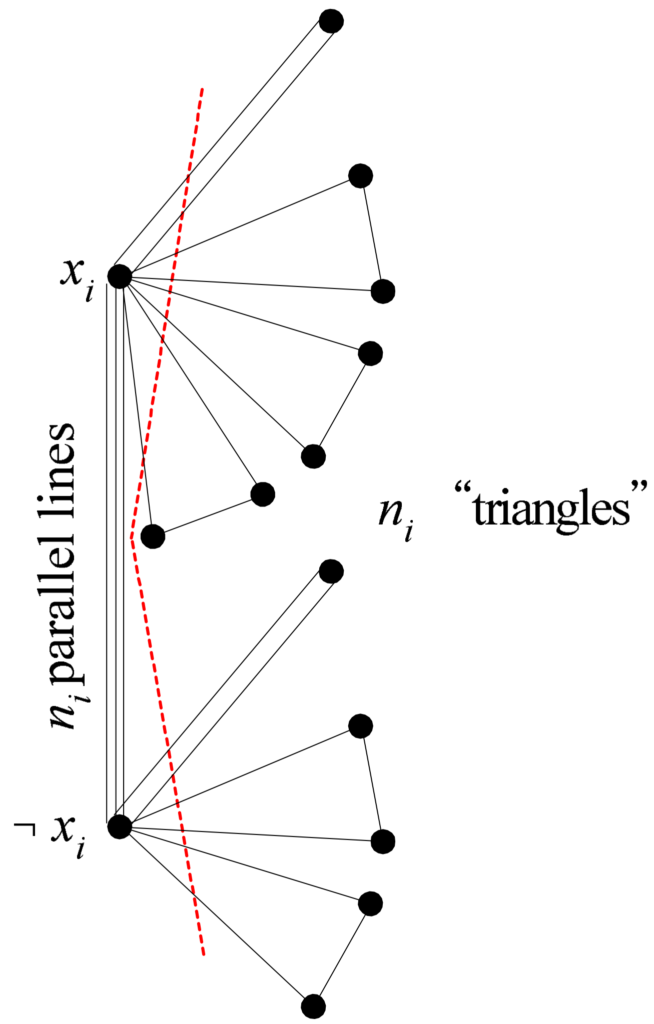
A Sample Construction (Cut Size Is 13)



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

The Proof

- 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.
- Then they together contribute at most $2n_i$ edges to the cut as they appear in at most n_i different clauses.



x_i
n_i parallel lines
$\neg\, x_i$
n_i “triangles”

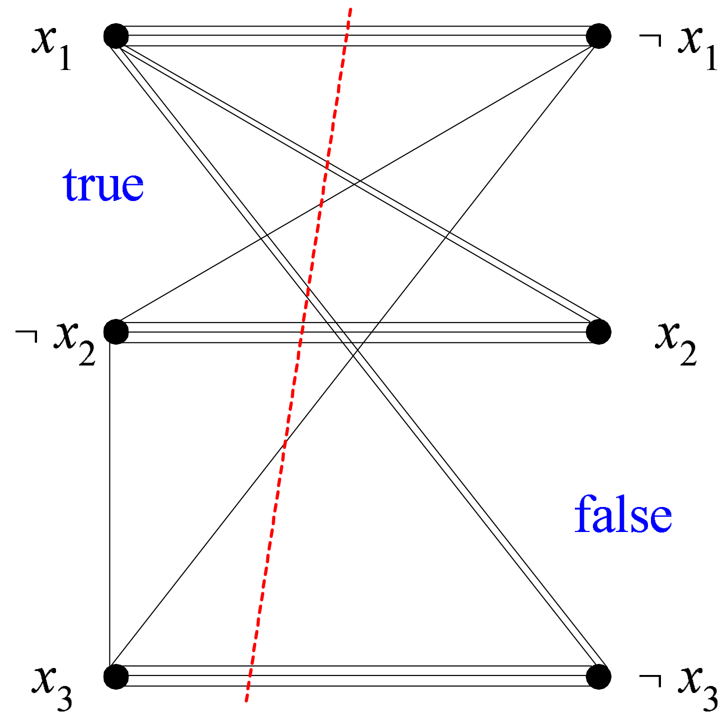
The Proof (continued)

- Changing the side of a literal contributing at most n_i to the cut 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 is $\sum_i n_i = 3m$.
  - The total number of literals is $3m$.

The Proof (concluded)

- The remaining $2m$ edges in the cut must come from the m triangles or parallel edges that correspond to the clauses.
- As each can contribute at most 2 to the cut, 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 New Cut (Cut Size Is 15)



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

©2002 Yuh-Dauh Lyuu, National Taiwan University

MAX BISECTION

- MAX CUT becomes MAX BISECTION if we require that $|S| = |V - S|$.
- It has many applications, especially in VLSI layout.
- Sometimes imposing additional restrictions makes a problem easier.
  - SAT to 2SAT.
- Other times, it makes the problem as hard or harder.
  - MIN CUT to BISECTION WIDTH.
  - LINEAR PROGRAMMING to INTEGER PROGRAMMING.

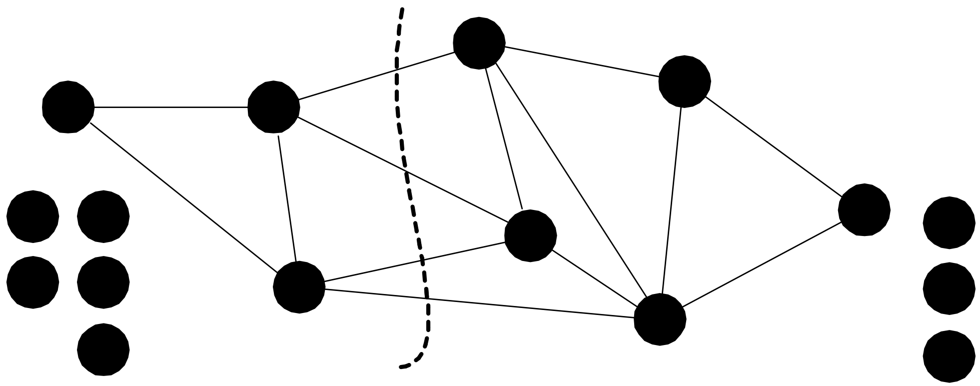
MAX BISECTION Is NP-Complete

- We shall reduce the *more general* MAX CUT to MAX BISECTION.
- Add $|V|$ **isolated nodes** to G to yield G'.
- G' has $2 \times |V|$ nodes.
- As the new nodes have no edges, moving them around contributes nothing to the cut.

©2002 Yuh-Dauh Lyuu, National Taiwan University

The Proof (concluded)

- Every cut $(S, V - S)$ of $G = (V, E)$ can be made into a bisection by appropriately allocating the new nodes between S and $V - S$.
- Hence each cut of G can be made a cut of G' of the same size, and vice versa.

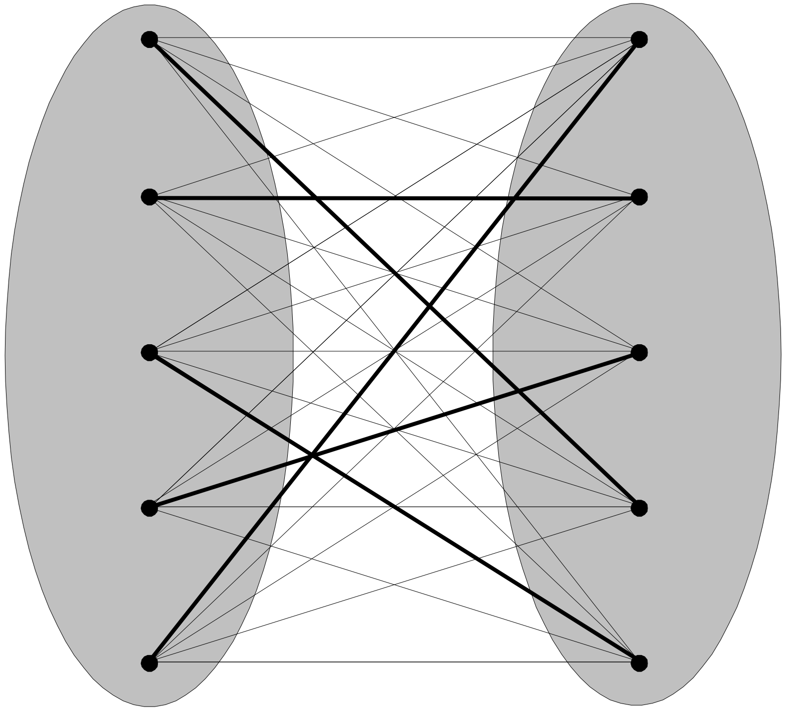


BISECTION WIDTH

- BISECTION WIDTH is like MAX BISECTION except that it asks if there is a bisection of size *at most* K (sort of MIN BISECTION).
- Unlike MIN CUT, BISECTION WIDTH remains NP-complete.
  - A graph $G = (V, E)$, where $|V| = 2n$, has a bisection of size K if and only if the complement of G has a bisection of size $n^2 - K$.

©2002 Yuh-Dauh Lyuu, National Taiwan University

Illustration



HAMILTONIAN PATH Is NP-Complete[a]

- Given an *undirected* graph, the question whether it has a Hamiltonian path is NP-complete.
- The "messy" reduction is from 3SAT.
- We skip the proof.

---

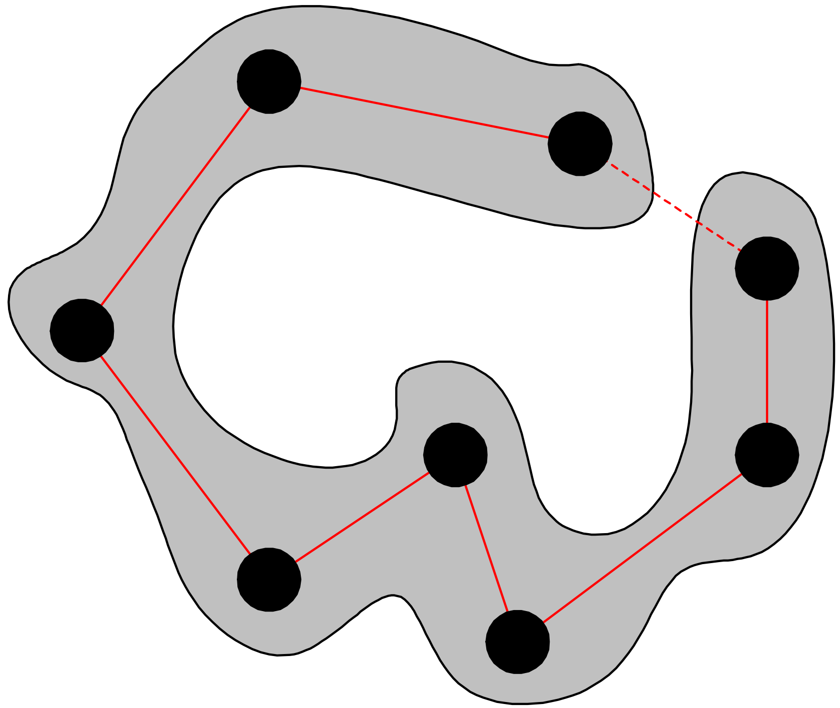
[a]Karp, 1972.

TSP (D) Is NP-Complete

Corollary 40 TSP (D) *is NP-complete.*

- Given a graph G with n nodes, define $d_{ij} = 1$ if $[\, i, j \,] \in G$ and $d_{ij} = 2$ if $[\, i, j \,] \notin G$.
- Set the budget $B = n + 1$.
- Note that if G has no Hamiltonian paths, then any tour must contain at least two edges with weight 2.
- The total cost is then at least $(n - 2) + 2 \cdot 2 = n + 2$.
- There is a tour of length B or less if and only if G has a Hamiltonian path.

Hamiltonian Path and TSP Tour



Graph Coloring

- k-COLORING asks if the nodes of a graph can be colored with k colors (or fewer) such that no two adjacent nodes have the same color.
- 2-COLORING is in P.
- 3-COLORING is NP-complete.
- Since 3-COLORING is a special case of k-COLORING for any $k \geq 4$, k-COLORING is NP-complete for $k \geq 3$.

3-COLORING Is NP-Complete[a]

- We will reduce NAESAT to 3-COLORING.
- We are given a set of clauses $C_1, C_2, \ldots, C_m$ each with 3 literals.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- We shall construct a graph G such that it can be colored with colors $\{0, 1, 2\}$ if and only if all the clauses can be NAE-satisfied.

[a] Karp, 1972.

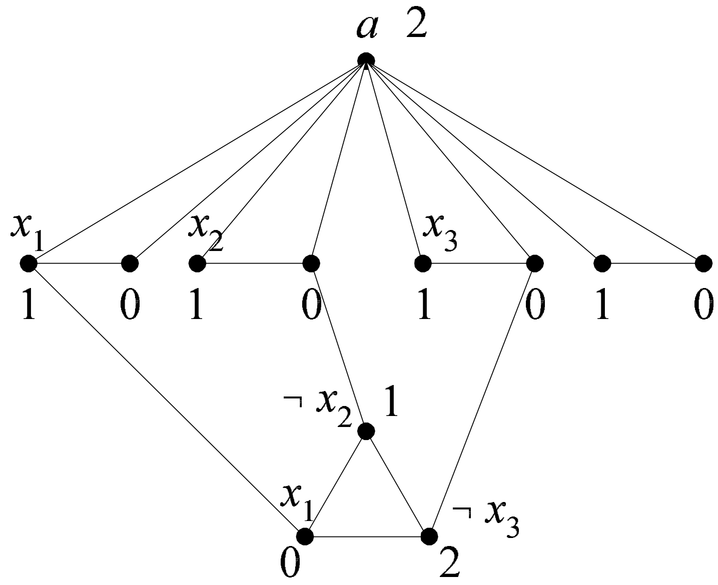
The Proof (continued)

- Every variable x_i is involved in a triangle $[\,a, x_i, \neg x_i\,]$ with a common node a.
- Each clause $C_i = (c_{i1} \vee c_{i2} \vee c_{i3})$ is also represented by a triangle

$$[\,c_{i1}, c_{i2}, c_{i3}\,].$$

- There is an edge between c_{ij} and the node that represents the jth literal of C_i.

Construction for $\cdots \land (x_1 \lor \neg x_2 \lor \neg x_3) \land \cdots$



The Proof (continued)

Suppose the graph is 3-colorable.

- Assume without loss of generality that node a takes the color 2, x_i takes the color 1, and $\neg x_i$ takes the color 0.
- A triangle must use all 3 colors.
- The clause triangle cannot be linked to nodes with all 1s or all 0s; otherwise, it cannot be colored with 3 colors.
- Treat 1 as true and 0 as false (it is consistent).
- Treat 2 as either true or false; it does not matter.
- As each clause triangle contains one color 1 and one color 0, the clauses are NAE-satisfied.

The Proof (concluded)

Suppose the clauses are NAE-satisfiable.

- Color node a with color 2.
- Color the nodes representing literals by their truth values (color 0 for false and color 1 for true).
- For each clause triangle:
  - Pick any two literals with opposite truth values and color the corresponding nodes with 0 if the literal is true and 1 if it is false.
  - Color the remaining node with color 2.