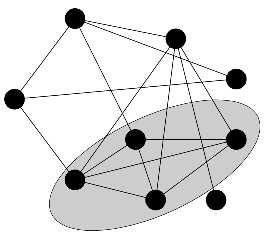
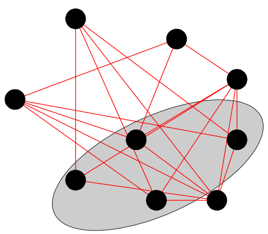
CLIQUE

- We are given an undirected graph G and a goal K.
- 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.

CLIQUE Is NP-Complete

Corollary 17 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 an independent set in $G \Leftrightarrow I$ is a clique in $\bar{G}$.

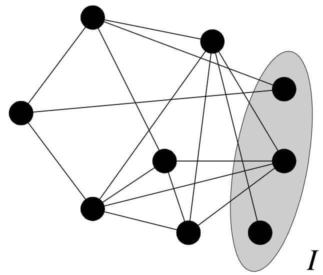
NODE COVER

- We are given an undirected graph G and a goal K.
- 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.

NODE COVER Is NP-Complete

Corollary 18 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

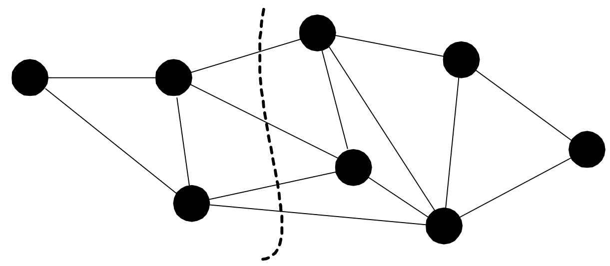
- A **cut** in an undirected graph $G = (V, E)$ is a partition of the nodes into two nonempty sets S and $V - S$.
- The size of a cut $(S, V - S)$ is the number of edges between S and $V - S$.
- MIN CUT $\in$ P by the maxflow algorithm.
- MAX CUT asks if there is a cut of size at least K.
  - K is part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

MIN CUT and MAX CUT (concluded)

- MAX CUT has applications in VLSI 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).

A Cut



MAX CUT Is NP-Complete[a]

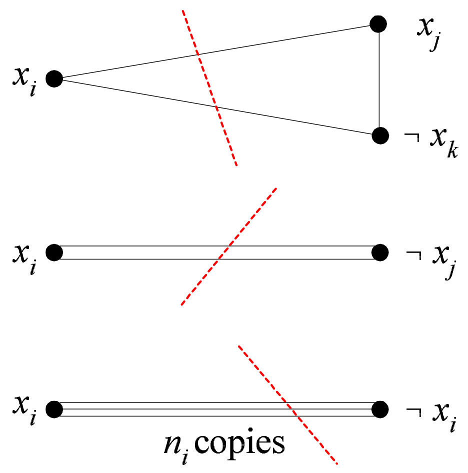
- We will reduce NAESAT to MAX CUT.
- Given an instance ϕ of 3SAT with m clauses, we shall construct a graph $G = (V, E)$ and a goal K such that:
  - There is a cut of size at least K if and only if ϕ is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes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

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

The Proof

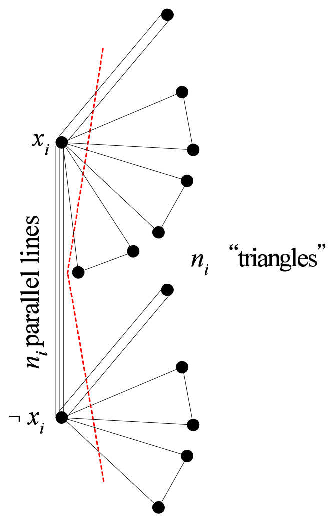
- Suppose ϕ's m clauses are $C_1, C_2, \ldots, C_m$.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- G has $2n$ nodes: $x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n$.
- Each clause with 3 distinct literals makes a triangle in G.
- For each clause with two identical literals, there 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.[a]

[a] Regardless of whether both x_i and $\neg x_i$ occur in ϕ.



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

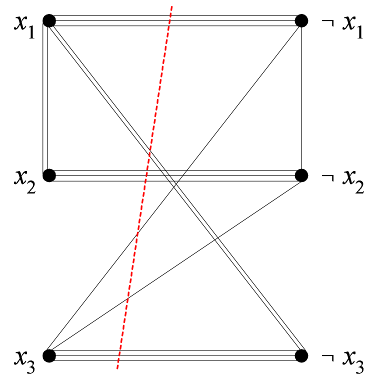


The Proof (continued)

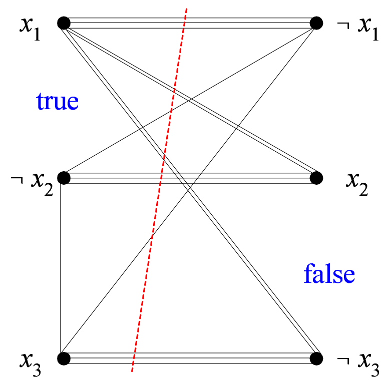
- Changing the side of a literal contributing at most n_i to the cut does not decrease the size of the cut.
- Hence we assume variables are separated from their negations.
- The total number of edges in the cut that join opposite literals is $\sum_i n_i = 3m$.
  - The total number of literals is $3m$.

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.



- $(x_1 \lor x_2 \lor x_2) \land (x_1 \lor \neg x_3 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$.
- The cut size is $13 < 5 \times 3 = 15$.



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

MAX BISECTION

- MAX CUT becomes MAX BISECTION if we require that $|S| = |V - S|$.
- It has many applications, especially in VLSI layout.

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.

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.