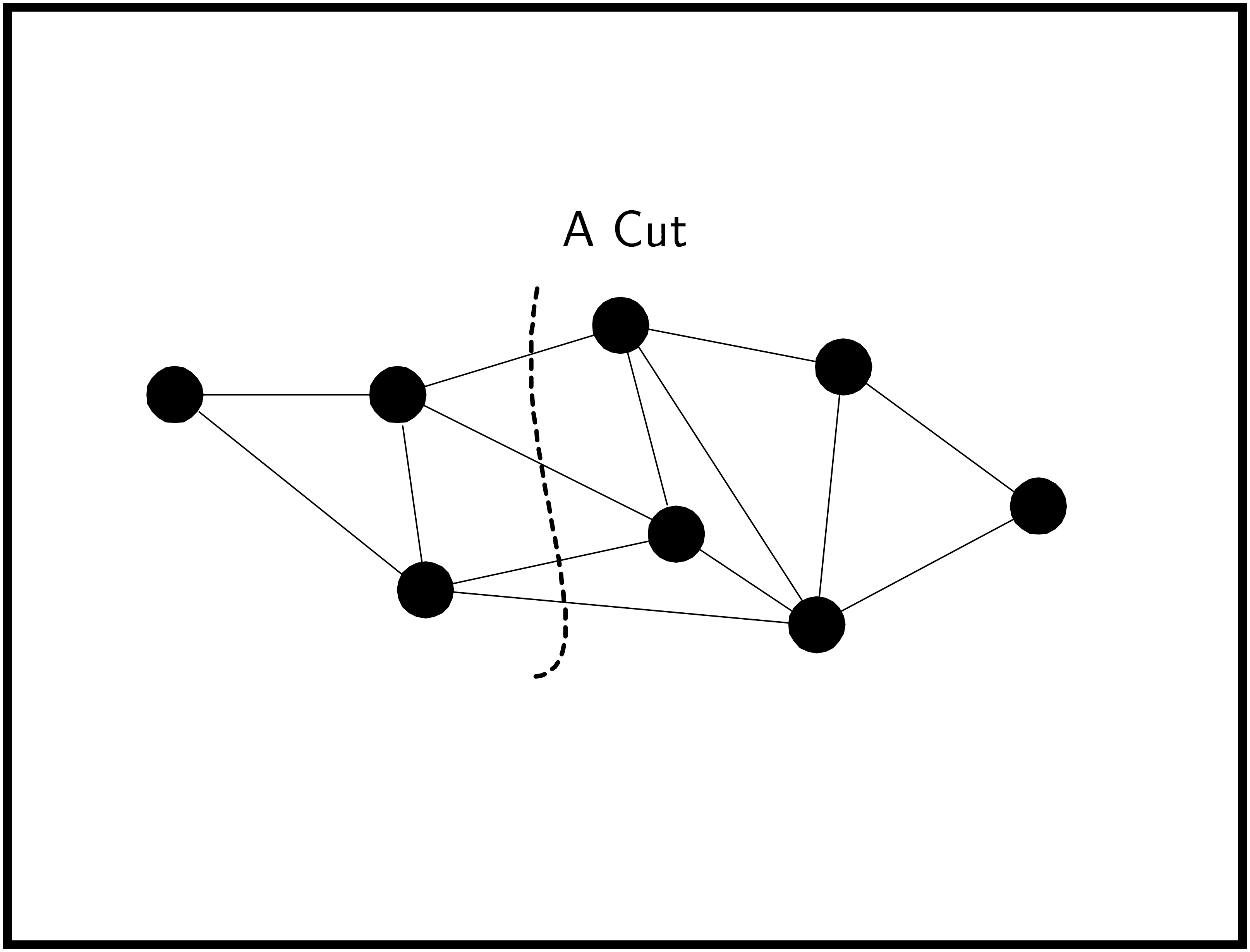
MIN CUT and MAX CUT

- 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 is in P.
- MAX CUT asks if there is a cut of size at least K.


A Cut

MAX CUT Is NP-Complete[a]

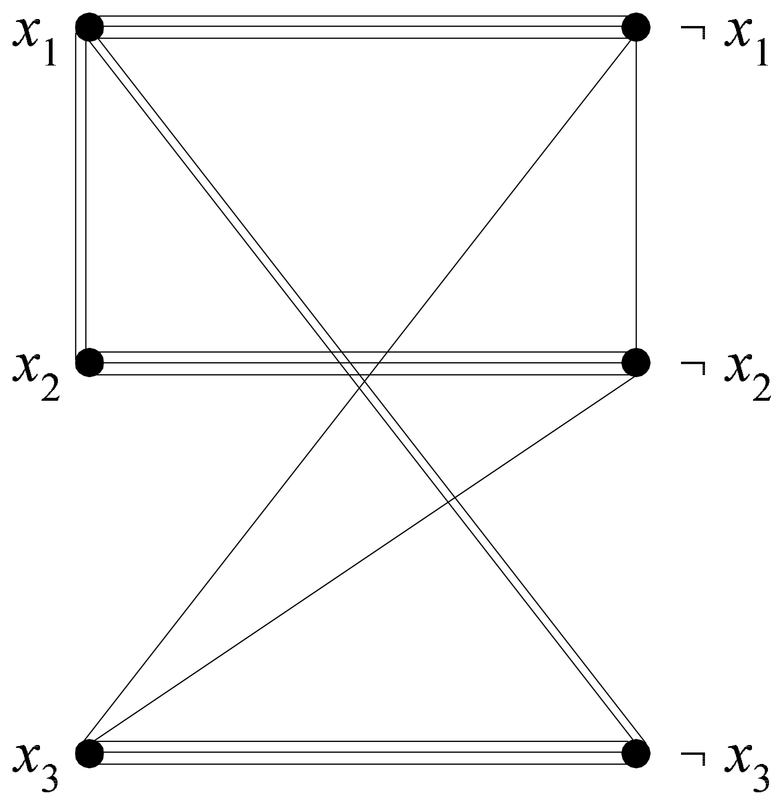
- We will reduce NAESAT to MAX CUT.
- Given an instance ϕ of 3SAT with m clauses, we shall construct a graph $G = (V, E)$ and a goal K such that there is a cut of size at least K if and only if ϕ is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes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

A Sample Construction



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

The Proof

- Set $K = 5m$.
- Suppose that 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 that 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 (continued)

- 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.

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.
- Other times, it makes the problem as hard or harder.

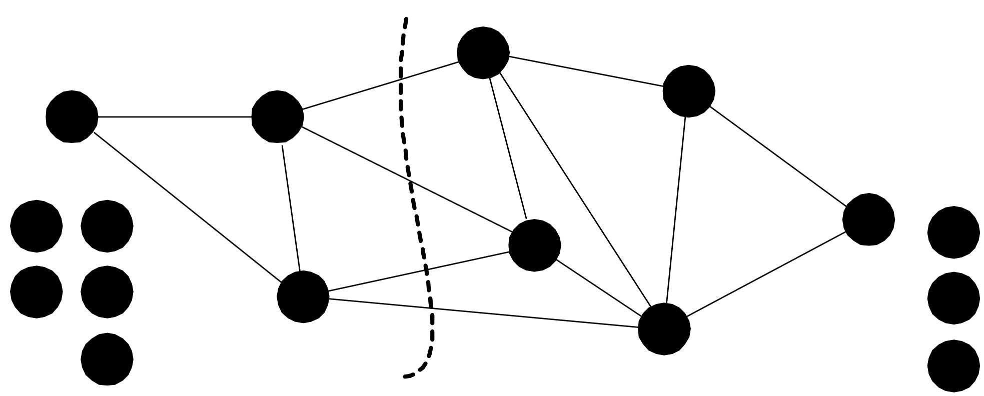
MAX BISECTION Is NP-Complete

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

The Proof (continued)

- 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 $n^2 - K$.
  - This trick will not work for MIN CUT vs. MAX CUT (why?).

HAMILTONIAN PATH Is NP-Complete[a]

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

---

[a] Karp, 1972.

TSP (D) Is NP-Complete

Corollary 35 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 $[\,i, j\,] \not\in G$.
- 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.

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]

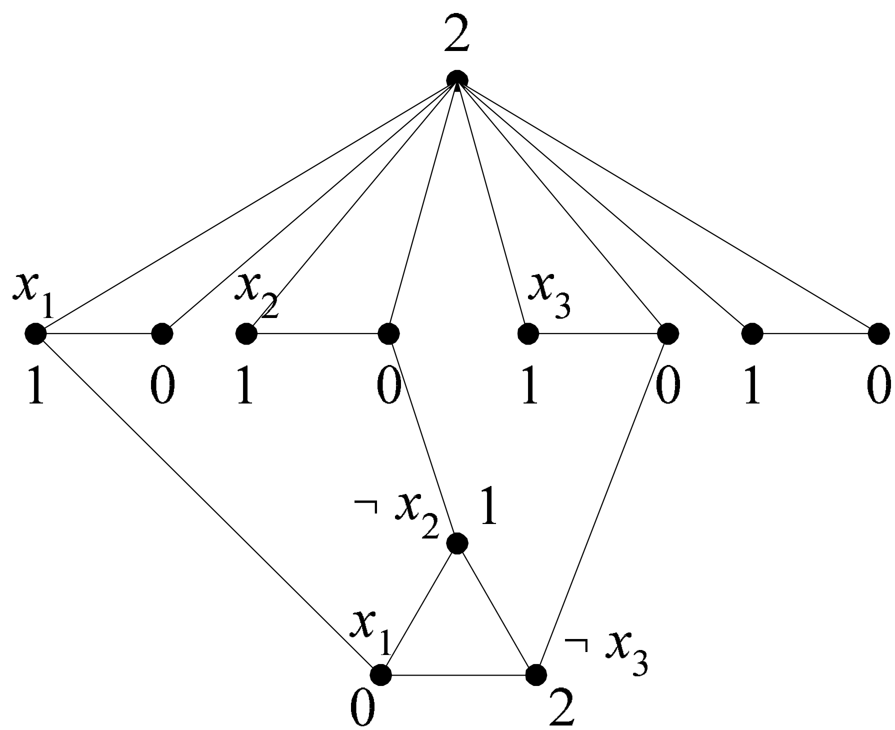
- We will reduce NAESAT to 3-COLORING.
- We are given a set of clauses $C_1, C_2, \ldots, C_m$ each with 3 literals.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- We shall construct a graph G such that it can be colored with colors $\{0, 1, 2\}$ if and only if all the clauses can be NAE-satisfied.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

[a]Karp, 1972.

The Proof (continued)

- Each clause $C_i = (c_{i1} \lor c_{i2} \lor c_{i3})$ 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.

A Clause in a Sample Construction



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

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.
- 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 (continued)

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.

TRIPARTITE MATCHING

- We are given three sets B, G, and H, each containing n elements.
- Let $T \subseteq B \times G \times H$ be a ternary relation.
- TRIPARTITE MATCHING asks if there is a set of n triples in T, no who of which have a component in common.
  - Each element in B is matched to a different element in G and different element in H.

Theorem 36 (Karp, 1972) TRIPARTITE MATCHING *is NP-complete.*

Related Problems

- We are given a family $F = \{S_1, S_2, \ldots, S_n\}$ of subsets of a finite set U and a budget B.
- SET COVERING asks if there exists a set of B sets in F whose union is U.
- SET PACKING asks if there are B disjoint sets in F.
- Assume $|U| = 3m$ for some $m \in \mathbb{N}$ and $|S_i| = 3$ for all i.
- EXACT COVER BY 3-SETS asks if there are m sets in F that are disjoint and have U as their union.

Corollary 37 SET COVERING, SET PACKING, *and* EXACT COVER BY 3-SETS *are all NP-complete.*

INTEGER PROGRAMMING

- INTEGER PROGRAMMING asks whether a given system of linear inequalities with integer coefficients has an integer solution.
- Many NP-complete problems can be expressed as an INTEGER PROGRAMMING problem.
  - SET COVERING can be expressed by the inequalities $Ax \geq \vec{1}$, $\sum_{i=1}^{n} x_i \leq B$, $0 \leq x_i \leq 1$, where
    * x_i is one if and only if S_i is in the cover.
    * A is the matrix whose columns are the bit vectors of the sets $S_1, S_2, \ldots$.
    * $\vec{1}$ is the vector of 1s.
- INTEGER PROGRAMMING is NP-complete [Papadimitriou, 1981].

The KNAPSACK Problem

- There are a set of n items.
- Item i has value $v_i \in \mathbb{Z}^+$ and weight $w_i \in \mathbb{Z}^+$.
- Given $K \in \mathbb{Z}^+$ and $W \in \mathbb{Z}^+$, KNAPSACK asks if there exists a subset $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq K$.
  - We want to achieve the maximum satisfaction within the budget.

KNAPSACK Is NP-Complete

- KNAPSACK is in NP: Guess an S and verify the constraints.
- We assume $v_i = w_i$ for all i and $K = W$.
- KNAPSACK becomes the problem if a subset of $\{w_1, w_2, \ldots, w_n\}$ adds up to exactly K.
  - Think of yourself as a radio DJ.
- We shall reduce EXACT COVER BY 3-SETS to it.
- We are given a family $F = \{S_1, S_2, \ldots, S_n\}$ of size-3 subsets of $U = \{1, 2, \ldots, 3m\}$.
- EXACT COVER BY 3-SETS asks if there are m disjoint sets in F that cover the set U.

The Proof (continued)

- Think of a set as a bit vector in $\{0, 1\}^{3m}$.
  - 001100010 means the set $\{3, 4, 8\}$, and 110010000 means the set $\{1, 2, 5\}$.
- A bit vector can also be considered as a binary *number*.
- Set union resembles addition.
  - $001100010 + 110010000 = 111110010$, which denotes the set $\{1, 2, 3, 4, 5, 8\}$, as desired.
- Trouble is there is *carry*.
  - $001100010 + 001110000 = 010010010$, which denotes the set $\{2, 5, 8\}$, not the desired $\{3, 4, 5, 8\}$.

The Proof (continued)

- To fix this problem, we only need to enlarge the base just enough so that there are no carries.
- Because there are n vectors in total, we change the base from 2 to $n + 1$.
- Finally, set

$$K = \sum_{j=0}^{3m-1} (n+1)^j,$$

which is $\overbrace{11 \cdots 1}^{3m}$ in base $n + 1$.

BIN PACKINGS

- We are given N positive integers $a_1, a_2, \ldots, a_N$, an integer C (the capacity), and an integer B (the number of bins).
- BIN PACKING asks if these numbers can be partitioned into B subsets, each of which has total sum at most C.
- Think of packing bags at the check-out counter.

Theorem 38 BIN PACKING *is NP-complete.*