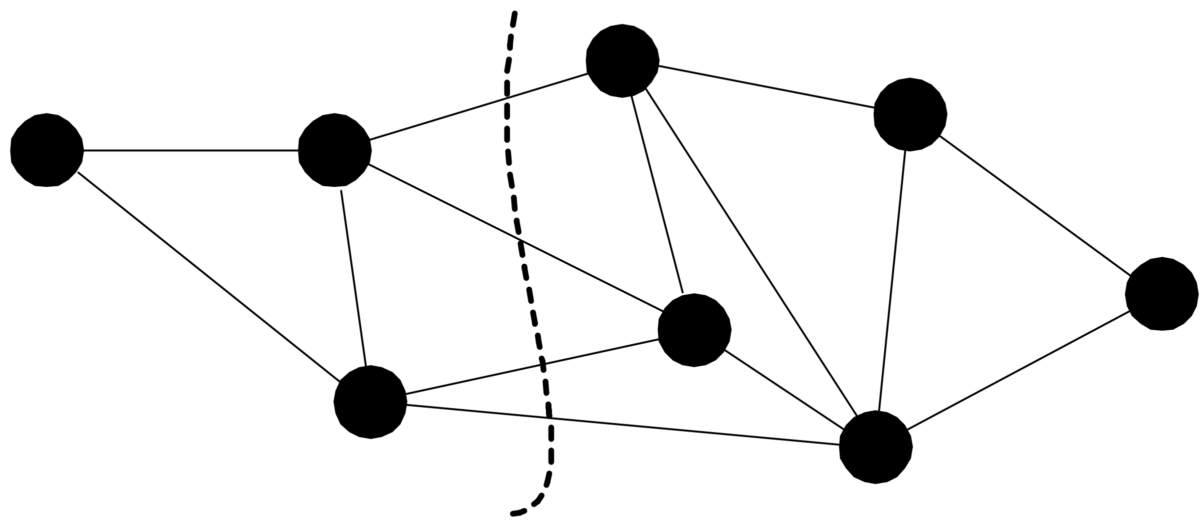
MIN CUT and MAX CUT

- 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.[a]
- MAX CUT asks if there is a cut of size at least K.
  - K is part of the input.

[a] 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.[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 a 3SAT formula ϕ 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]Karp (1972); Garey, Johnson, and Stockmeyer (1976).

The Proof

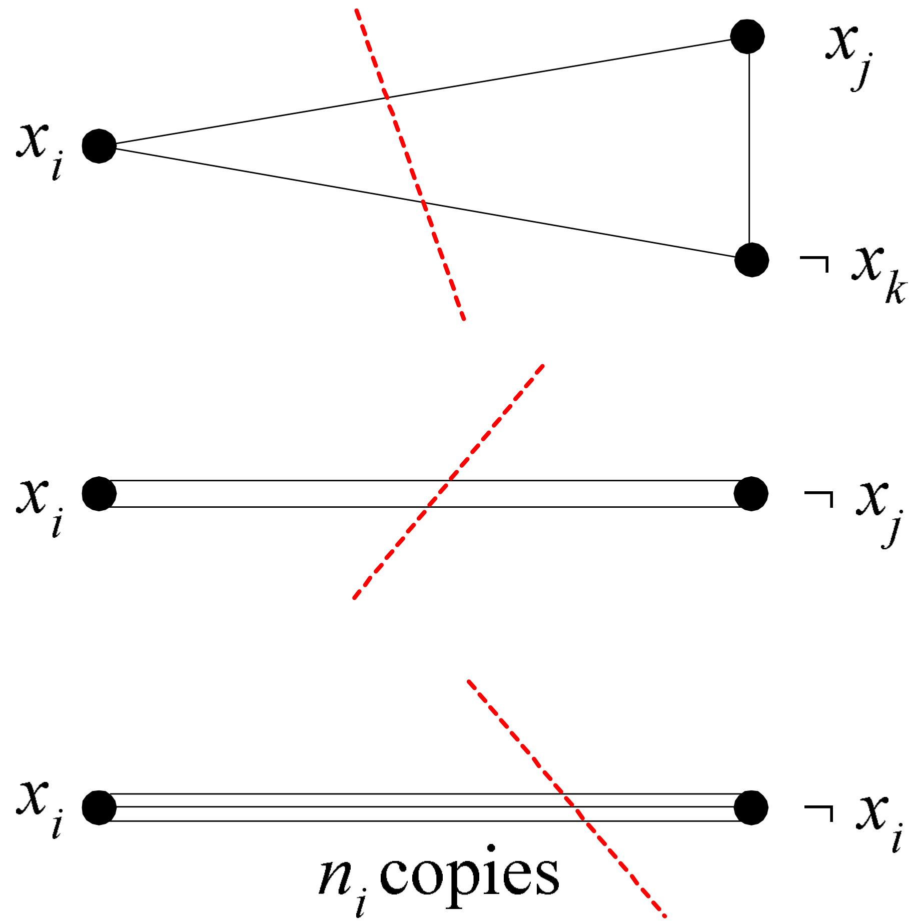
- Suppose ϕ's m clauses are $C_1, C_2, \ldots, C_m$.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- G has $2n$ nodes: $x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n$.
- Each clause with 3 distinct literals makes a triangle in G.
- For each clause with two identical literals, there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

The Proof (continued)

- No need to consider clauses with one literal (why?).
- No need to consider clauses containing two opposite literals x_i and $\neg x_i$ (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.
- Note that

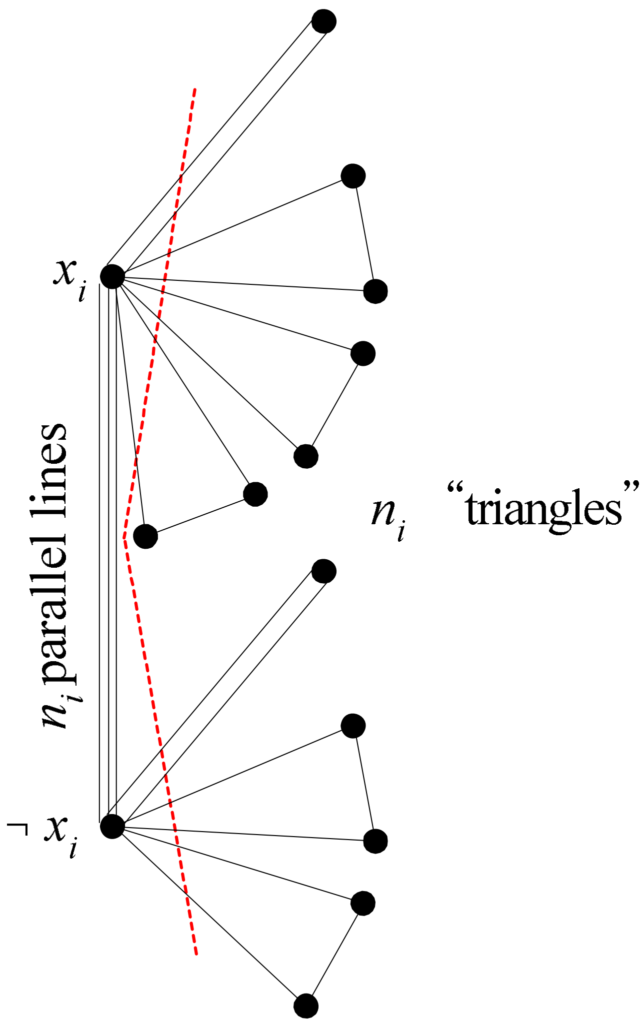
$$\sum_{i=1}^{n} n_i = 3m.$$

  - The summation is simply the total number of literals.


x_j
x_i
$\neg x_k$
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 some 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.


x_i
n_i parallel lines
n_i "triangles"
$\neg\, x_i$

The Proof (continued)

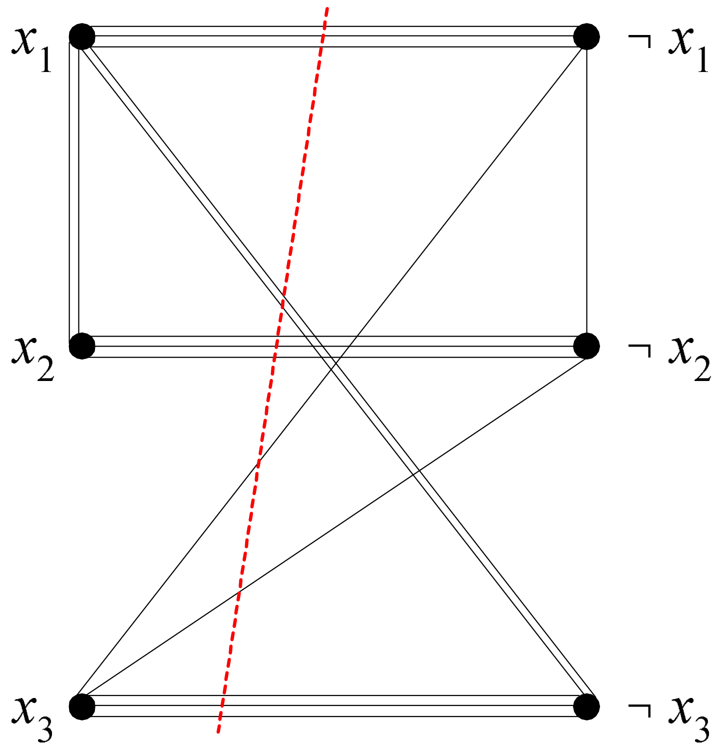
- Either x_i or $\neg x_i$ contributes at most n_i to the cut by the pigeonhole principle.
- Changing the side of that literal does *not decrease* the size of the cut.
- Hence we assume variables are separated from their negations.
- The total number of edges in the cut that join opposite literals x_i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

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.[a]
- 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.

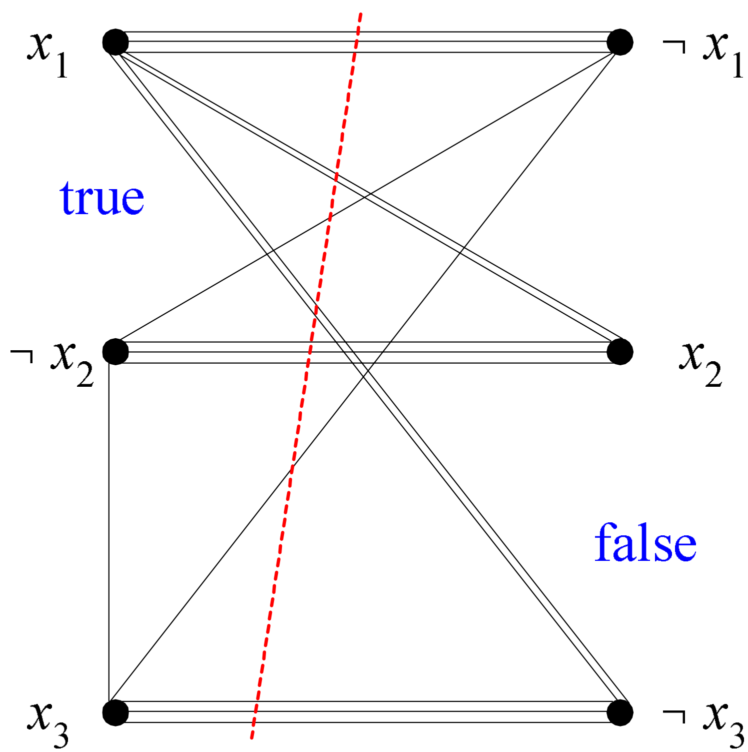
[a] So $K = 5m$.

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



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

This Cut 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.[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.

MAX BISECTION

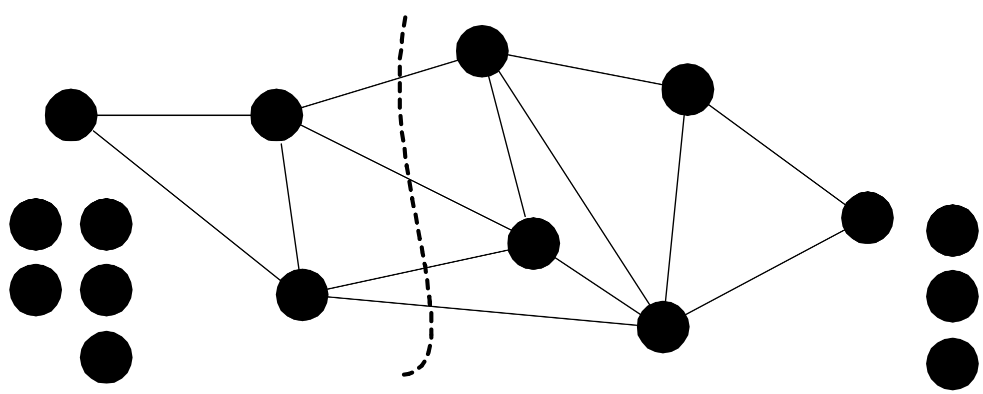
- MAX CUT becomes MAX BISECTION if we require that $|\,S\,| = |\,V - S\,|$.
- It has many applications, especially in VLSI layout.

MAX BISECTION Is NP-Complete

- We shall reduce the more general MAX CUT to MAX BISECTION.
- Add $|V| = n$ **isolated nodes** to G to yield G'.
- G' has $2n$ nodes.
- G''s goal K is identical to G's
  – As the new nodes have no edges, they contribute 0 to the cut.
- This completes the reduction.

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 is NP-complete.
- We reduce MAX BISECTION to BISECTION WIDTH.
- Given a graph $G = (V, E)$, where $|V|$ is even, we generate the complement of G.
- Given a goal of K, we generate a goal of $n^2 - K$.[a]

[a] $|V| = 2n$.

The Proof (concluded)

- To show the reduction works, simply notice the following easily verifiable claims.
  - A graph $G = (V, E)$, where $|V| = 2n$, has a bisection of size K if and only if the complement[a] of G has a bisection of size $n^2 - K$.
  - So G has a bisection of size $\geq K$ if and only if its complement has a bisection of size $\leq n^2 - K$.

---

[a]Recall p. 374.

HAMILTONIAN PATH Is NP-Complete[a]

Theorem 45 *Given an* undirected *graph, the question whether it has a Hamiltonian path is NP-complete.*

---

[a]Karp (1972).

A Hamiltonian Path at IKEA, Covina, California?

TSP (D) Is NP-Complete

Corollary 46 TSP (D) *is NP-complete.*

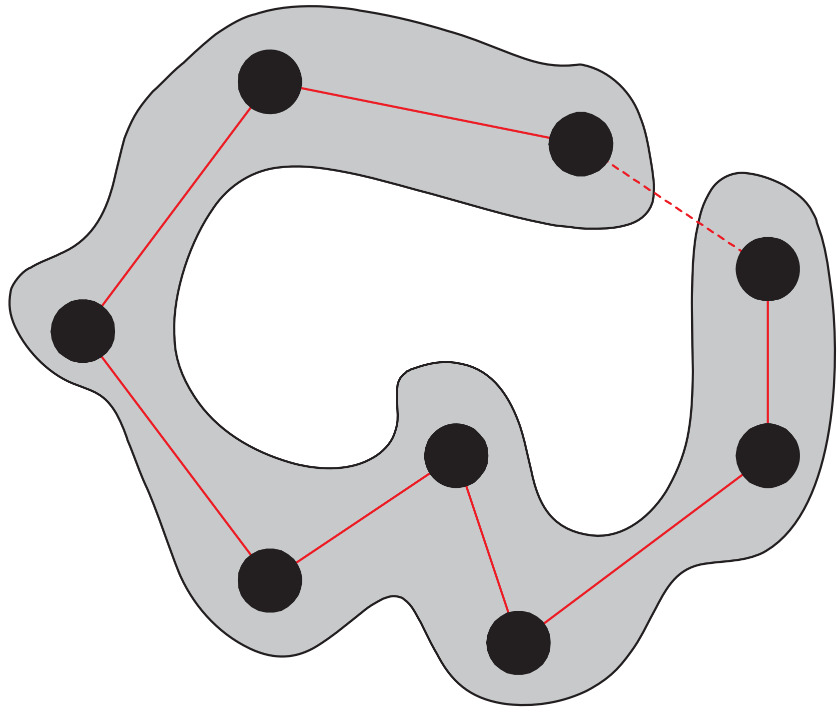
- Consider a graph G with n nodes.
- Create a weighted complete graph G' with the same nodes as G.
- Set $d_{ij} = 1$ on G' if $[\, i, j\,] \in G$ and $d_{ij} = 2$ on G' if $[\, i, j\,] \notin G$.
  - Note that G' is a complete graph.
- Set the budget $B = n + 1$.
- This completes the reduction.

TSP (D) Is NP-Complete (continued)

- Suppose G' has a tour of distance at most $n+1$.[a]
- Then that tour on G' must contain at most one edge with weight 2.
- If a tour on G' contains one edge with weight 2, remove that edge to arrive at a Hamiltonian path for G.
- Suppose a tour on G' contains no edge with weight 2.
- Remove any edge to arrive at a Hamiltonian path for G.

---

[a] A tour is a cycle, not a path.



TSP (D) Is NP-Complete (concluded)

- On the other hand, suppose G has a Hamiltonian path.
- There is a tour on G' containing at most one edge with weight 2.
  - Start with a Hamiltonian path and then close the loop.
- The total cost is then at most $(n-1)+2=n+1=B$.
- We conclude that there is a tour of length B or less on G' if and only if G has a Hamiltonian path.

Random TSP

- Suppose each distance d_{ij} is picked uniformly and independently from the interval $[\,0, 1\,]$.
- It is known that the total distance of the shortest tour has a mean value of $\beta\sqrt{n}$ for some positive β.
- In fact, the total distance of the shortest tour deviates from the mean by more than t with probability at most $e^{-t^2/(4n)}$![a]

[a] Dubhashi and Panconesi (2012).

Graph Coloring

- k-COLORING: Can the nodes of a graph be colored with $\leq k$ colors such that no two adjacent nodes have the same color?[a]
- 2-COLORING is in P (why?).
- But 3-COLORING is NP-complete (see next page).
- k-COLORING is NP-complete for $k \geq 3$ (why?).
- EXACT-k-COLORING asks if the nodes of a graph can be colored using exactly k colors.
- It remains NP-complete for $k \geq 3$ (why?).

[a] k is *not* part of the input; k is part of the problem statement.

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 that can be colored with colors $\{ 0, 1, 2 \}$ if and only if all the clauses can be NAE-satisfied.

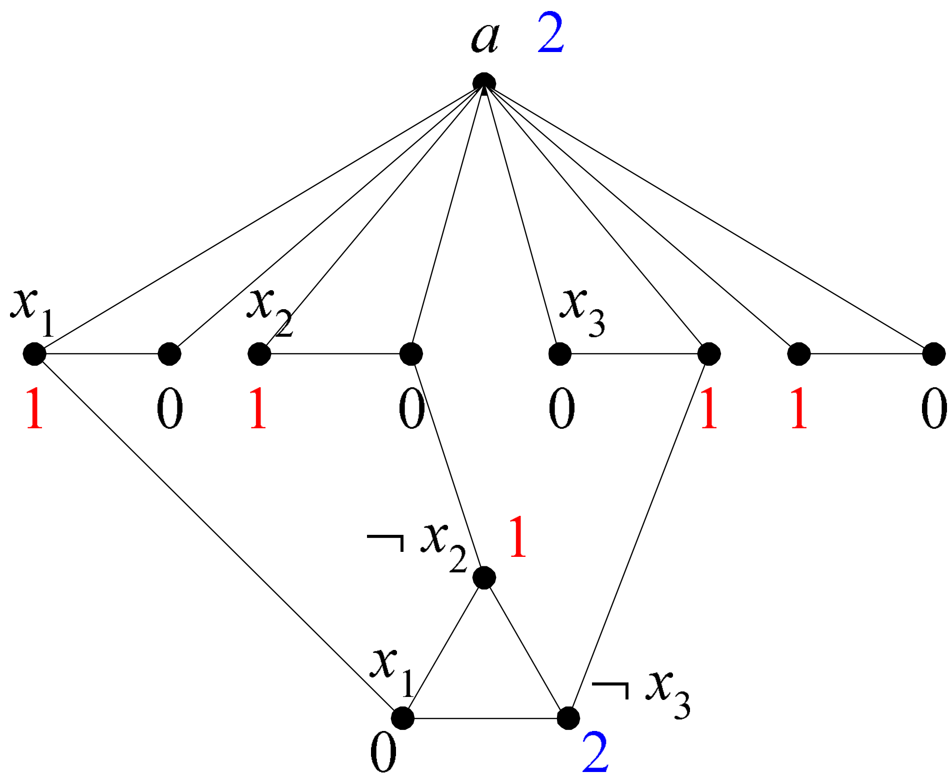
[a]Karp (1972).

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
$$[c_{i1}, c_{i2}, c_{i3}].$$
  - Node c_{ij} and a node in an a-triangle $[a, x_k, \neg x_k]$ with the same label represent *distinct* nodes.
- There is an edge between c_{ij} and the node that represents the jth literal of C_i.[a]

[a]Alternative proof: There is an edge between $\neg c_{ij}$ and the node that represents the jth literal of C_i. Contributed by Mr. Ren-Shuo Liu (`D98922016`) on October 27, 2009.

Construction for $\cdots \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \cdots$



The Proof (continued)

Suppose the graph is 3-colorable.

- Assume without loss of generality that node a takes the color 2.
- A triangle must use up all 3 colors.
- As a result, one of x_i and $\neg x_i$ must take the color 0 and the other 1.

The Proof (continued)

- Treat 1 as `true` and 0 as `false`.[a]
  - We are dealing with the a-triangles here, not the clause triangles yet.
- The resulting truth assignment is clearly contradiction free.
- As each clause triangle contains one color 1 and one color 0, the clauses are NAE-satisfied.

[a] The opposite also works.

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`).
  - We are dealing with the a-triangles here, not the clause triangles.

The Proof (continued)

- For each clause triangle:
  - Pick any two literals with opposite truth values.[a]
  - Color the corresponding nodes with 0 if the literal is `true` and 1 if it is `false`.
  - Color the remaining node with color 2.

[a] Break ties arbitrarily.

The Proof (concluded)

- The coloring is legitimate.
  - If literal w of a clause triangle has color 2, then its color will never be an issue.
  - If literal w of a clause triangle has color 1, then it must be connected up to literal $\bar{w}$ with color 0.
  - If literal w of a clause triangle has color 0, then it must be connected up to literal $\bar{w}$ with color 1.

Algorithms for 3-COLORING and the Chromatic Number $\chi(G)$

- Assume G is 3-colorable.
- There is a classic algorithm that finds a 3-coloring in time $O(3^{n/3}) = 1.4422^n$.[a]
- It can be improved to $O(1.3289^n)$.[b]

---

[a] Lawler (1976).
[b] Beigel and Eppstein (2000).

Algorithms for 3-COLORING and the Chromatic Number $\chi(G)$ (concluded)

- The **chromatic number** $\chi(G)$ is the smallest number of colors needed to color a graph G.
- There is an algorithm to find $\chi(G)$ in time $O((4/3)^{n/3}) = 2.4422^n$.[a]
- It can be improved to $O((4/3 + 3^{4/3}/4)^n) = O(2.4150^n)$[b] and $2^n n^{O(1)}$.[c]
- Computing $\chi(G)$ cannot be easier than 3-COLORING.[d]

---

[a] Lawler (1976).

[b] Eppstein (2003).

[c] Koivisto (2006).

[d] Contributed by Mr. Ching-Hua Yu (`D00921025`) on October 30, 2012.

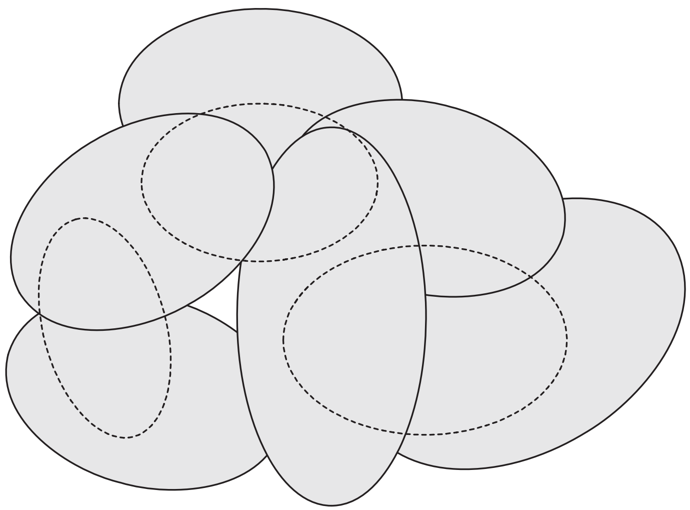
TRIPARTITE MATCHING

- 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, none of which has a component in common.
  - Each element in B is matched to a different element in G and different element in H.

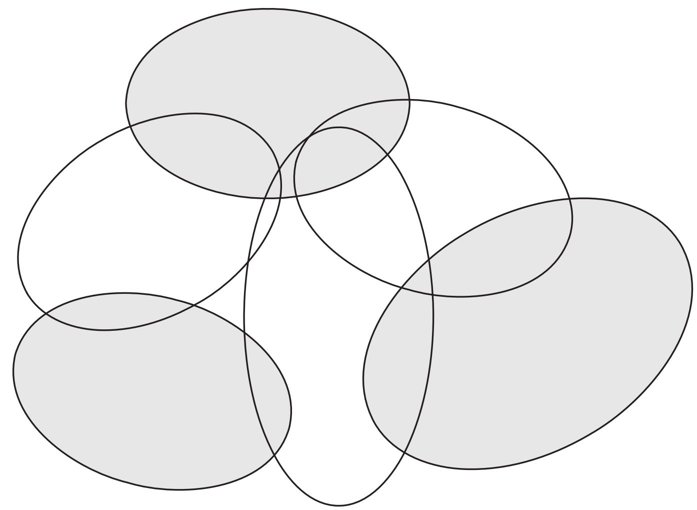
Theorem 47 (Karp (1972)) TRIPARTITE MATCHING *is NP-complete.*

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 (so have U as their union).



SET COVERING



SET PACKING

Related Problems (concluded)

Corollary 48 (Karp (1972)) SET COVERING, SET PACKING, *and* EXACT COVER BY 3-SETS *are all NP-complete.*

- SET COVERING is used to prove that the influence maximization problem in social networks is NP-complete.[a]

[a]Kempe, Kleinberg, and Tardos (2003).

KNAPSACK

- There is a set of n items.
- Item i has value $v_i \in \mathbb{Z}^+$ and weight $w_i \in \mathbb{Z}^+$.
- We are given $K \in \mathbb{Z}^+$ and $W \in \mathbb{Z}^+$.
- KNAPSACK asks if there exists a subset

$$I \subseteq \{1, 2, \ldots, n\}$$

  such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i \geq K$.
  - We want to achieve the maximum satisfaction within the budget.

KNAPSACK Is NP-Complete[a]

- KNAPSACK $\in$ NP: Guess an I and check the constraints.
- We shall reduce EXACT COVER BY 3-SETS to KNAPSACK, in which $v_i = w_i$ for all i and $K = W$.
- The simplified KNAPSACK now asks if a subset of $v_1, v_2, \ldots, v_n$ adds up to exactly K.[b]
  - Picture yourself as a radio DJ.

---

[a]Karp (1972).

[b]This problem is called SUBSET SUM.

The Proof (continued)

- The primary differences between the two problems are:[a]
  - Sets vs. numbers.
  - Union vs. addition.
- 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.

[a] Thanks to a lively class discussion on November 16, 2010.

The Proof (continued)

- Think of a set as a bit vector in $\{0, 1\}^{3m}$.
  - Assume $m = 3$.
  - `110010000` means the set $\{1, 2, 5\}$.
  - `001100010` means the set $\{3, 4, 8\}$.
- Assume there are $n = 5$ size-3 subsets in F.
- Our goal is

$$\overbrace{\mathtt{1}\,\mathtt{1}\cdots\mathtt{1}}^{3m}.$$

The Proof (continued)

- A bit vector can also be seen as a binary *number*.
- Set union resembles addition:

$$\begin{array}{rr} & \texttt{001100010} \\ + & \texttt{110010000} \\ \hline & \texttt{111110010} \end{array}$$

which denotes the set $\{1, 2, 3, 4, 5, 8\}$, as desired.

The Proof (continued)

- Trouble occurs when there is *carry*:

$$\begin{array}{rr} & \texttt{010000000} \\ + & \texttt{010000000} \\ \hline & \texttt{100000000} \end{array}$$

which denotes the wrong set $\{1\}$, not the correct $\{2\}$.

The Proof (continued)

- Or consider

$$
\begin{array}{rr}
 & \texttt{001100010} \\
+ & \texttt{001110000} \\
\hline
 & \texttt{011010010}
\end{array}
$$

which denotes the set $\{2, 3, 5, 8\}$, not the correct $\{3, 4, 5, 8\}$.[a]

[a] Corrected by Mr. Chihwei Lin (D97922003) on January 21, 2010.

The Proof (continued)

- Carry may also lead to a situation where we obtain our solution $\texttt{1}\,\texttt{1}\cdots\texttt{1}$ with more than m sets in F.
- For example,

$$
\begin{array}{rr}
 & \texttt{000100010} \\
 & \texttt{001110000} \\
 & \texttt{101100000} \\
+ & \texttt{000001101} \\
\hline
 & \texttt{111111111}
\end{array}
$$

- But the correct answer, $\{1,3,4,5,6,7,8,9\}$, is *not* an exact cover.

The Proof (continued)

- And it uses 4 sets instead of the required $m = 3$.[a]
- To fix this problem, we enlarge the base just enough so that there are no carries.[b]
- Because there are n vectors in total, we change the base from 2 to $n + 1$.

---

[a] Thanks to a lively class discussion on November 20, 2002.

[b] You cannot map $\cup$ to $\vee$ because KNAPSACK requires $+$ not $\vee$!

The Proof (continued)

- Set v_i to be the integer corresponding to the bit vector encoding S_i in base $n+1$:

$$v_i = \sum_{j \in S_i} 1 \times (n+1)^{3m-j} \tag{3}$$

- Set

$$K = \sum_{j=0}^{3m-1} 1 \times (n+1)^j = \overbrace{1\,1 \cdots 1}^{3m} \quad \text{(base } n+1\text{)}.$$

- Now in base $n+1$, if there is a set S such that $\sum_{i \in S} v_i = \overbrace{1\,1 \cdots 1}^{3m}$, then every position must be contributed by exactly one v_i and $|S| = m$.

The Proof (continued)

- For example, the case on p. 423 becomes

$$
\begin{array}{rr}
 & 000100010 \\
 & 001110000 \\
 & 101100000 \\
+ & 000001101 \\
\hline
 & 102311111
\end{array}
$$

in base $n + 1 = 6$.

- As desired, it no longer meets the goal.

The Proof (continued)

- Suppose F admits an exact cover, say $\{S_1, S_2, \ldots, S_m\}$.
- Then picking $I = \{1, 2, \ldots, m\}$ clearly results in

$$v_1 + v_2 + \cdots + v_m = \overbrace{1\,1 \cdots 1}^{3m}.$$

- It is important to note that the meaning of addition (+) is independent of the base.[a]
  - It is just regular addition.
  - But an S_i may give rise to different integers v_i in Eq. (3) on p. 425 under different bases.

[a] Contributed by Mr. Kuan-Yu Chen (R92922047) on November 3, 2004.

The Proof (concluded)

- On the other hand, suppose there exists an I such that

$$\sum_{i \in I} v_i = \overbrace{1\,1 \cdots 1}^{3m}$$

in base $n+1$.

- The no-carry property implies that $|\,I\,| = m$ and

$$\{S_i : i \in I\}$$

is an exact cover.

An Example

- Let $m = 3$, $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and

$$\begin{aligned} S_1 &= \{1, 3, 4\}, \\ S_2 &= \{2, 3, 4\}, \\ S_3 &= \{2, 5, 6\}, \\ S_4 &= \{6, 7, 8\}, \\ S_5 &= \{7, 8, 9\}. \end{aligned}$$

- Note that $n = 5$, as there are 5 S_i's.

An Example (continued)

- Our reduction produces

$$
\begin{aligned}
K &= \sum_{j=0}^{3\times 3-1} 6^j = \overbrace{1\,1\cdots 1}^{3\times 3}{}_6 = 2015539_{10}, \\
v_1 &= \texttt{101100000} = 1734048, \\
v_2 &= \texttt{011100000} = 334368, \\
v_3 &= \texttt{010011000} = 281448, \\
v_4 &= \texttt{000001110} = 258, \\
v_5 &= \texttt{000000111} = 43.
\end{aligned}
$$

An Example (concluded)

- Note $v_1 + v_3 + v_5 = K$ because

$$\begin{array}{rr} & 101100000 \\ & 010011000 \\ + & 000000111 \\ \hline & 111111111 \end{array}$$

- Indeed,

$$S_1 \cup S_3 \cup S_5 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$$

an exact cover by 3-sets.

BIN PACKING

- 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 49 BIN PACKING *is NP-complete.*

BIN PACKING (concluded)

- But suppose $a_1, a_2, \ldots, a_N$ are randomly distributed between 0 and 1.
- Let B be the smallest number of unit-capacity bins capable of holding them.
- Then B can deviate from its average by more than t with probability at most $2e^{-2t^2/N}$.[a]

---

[a] Dubhashi and Panconesi (2012).