MAX BISECTION

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

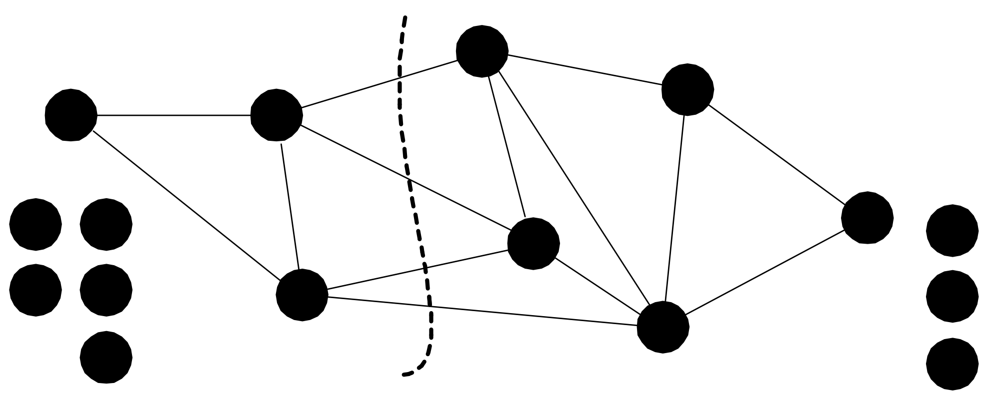
MAX BISECTION Is NP-Complete

- 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 nothing 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 $n^2 - K$.

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 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 $\leq n^2 - K$.

HAMILTONIAN PATH Is NP-Complete[a]

Theorem 42 *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?

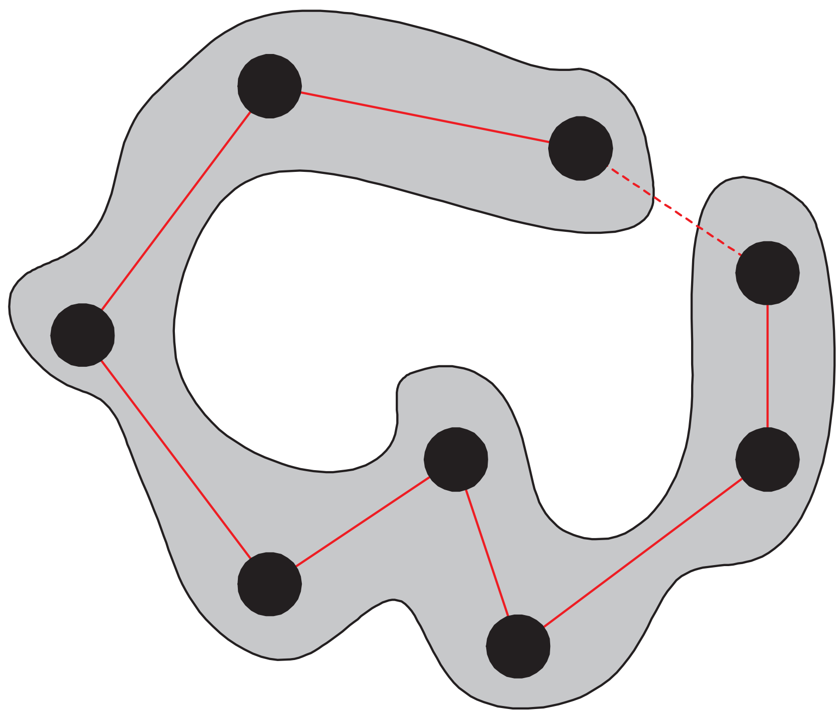
TSP (D) Is NP-Complete

Corollary 43 TSP (D) *is NP-complete.*

- Consider a graph G with n nodes.
- Create a weighted complete graph G' with the same nodes as from G follows.
- 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 $n + 1$.
- Then that tour on G' must contain at most one edge with weight 2.
- If a tour on G' contains 1 edge with weight 2, remove that edge to arrive at a Hamiltonian path for G.
- If, on the other hand, a tour on G' contains no edge with weight 2.
- Remove any edge to arrive at a Hamiltonian path for G.



TSP (D) Is NP-Complete (concluded)

- The total cost is then at least $(n-2)+2\cdot 2 = n+2 > B$.
- On the other hand, suppose G has Hamiltonian paths.
- Then there is a tour on G' containing at most one edge with weight 2.
- 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.

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?
- 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?).

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.

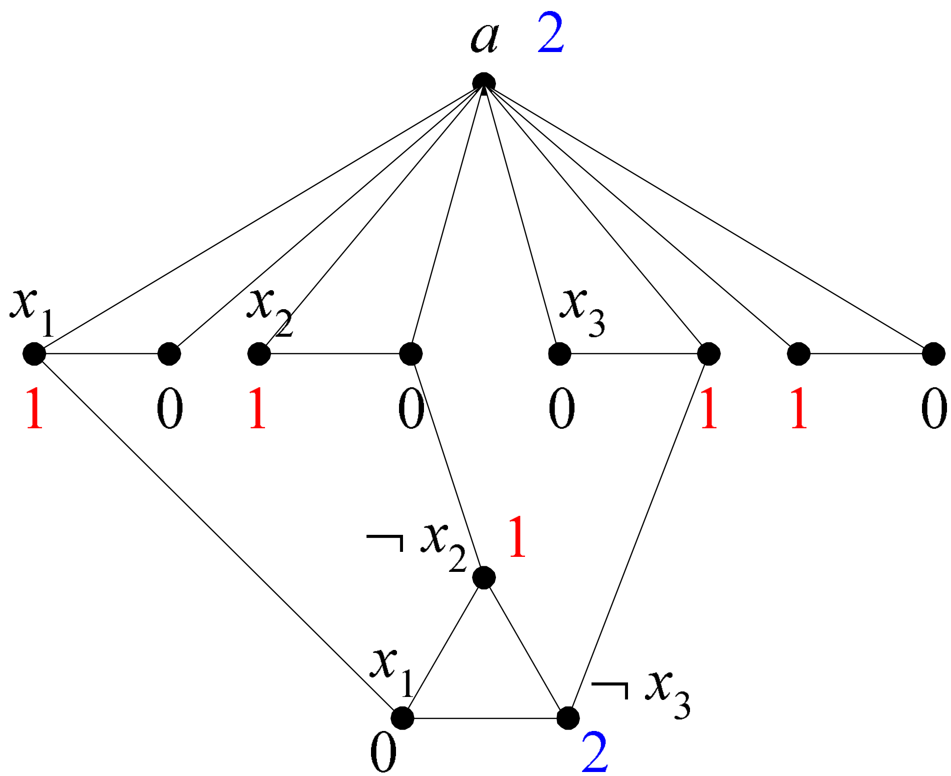
[a] Karp (1972).

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

[a]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 were dealing only with those triangles with the "a" node, not the clause triangles.
- 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 were dealing only with those triangles with the "a" node, not the clause triangles.

The Proof (continued)

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

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 w with color 0.
  - If literal w of a clause triangle has color 0, then it must be connected up to literal w with color 1.

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

- Assume G is 3-colorable.
- There is an algorithm to find a 3-coloring in time $O(3^{n/3}) = 1.4422^n$.[a]
- It has been improved to $O(1.3289^n)$.[b]
- There is an algorithm to find $\chi(G)$ in time $O((4/3)^{n/3}) = 2.4422^n$.[c]
- It can be improved to $O((4/3 + 3^{4/3}/4)^n) = O(2.4150^n)$.[d]

---

[a] Lawler (1976).
[b] Beigel and Eppstein (2000).
[c] Lawler (1976).
[d] Eppstein (2003).

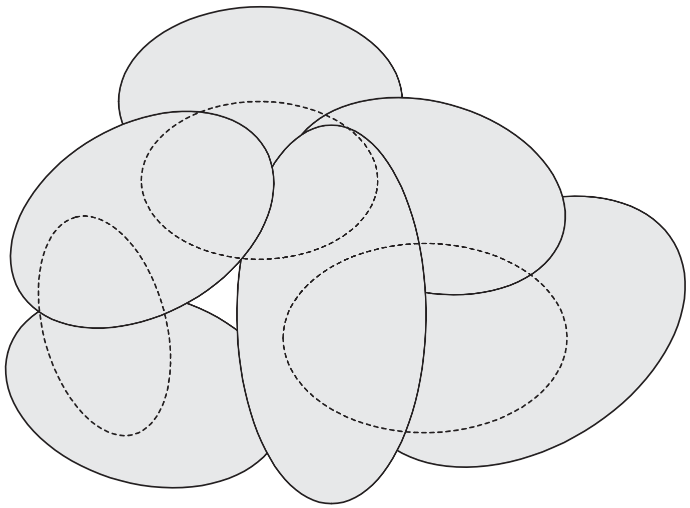
TRIPARTITE MATCHING

- 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.

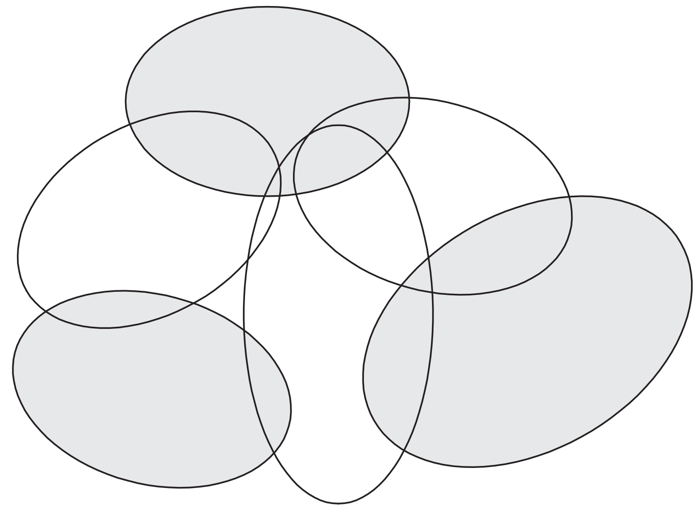
Theorem 44 (Karp (1972)) TRIPARTITE MATCHING *is NP-complete.*

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.



SET COVERING



SET PACKING

Related Problems (concluded)

Corollary 45 SET COVERING, SET PACKING, *and* EXACT COVER BY 3-SETS *are all NP-complete.*