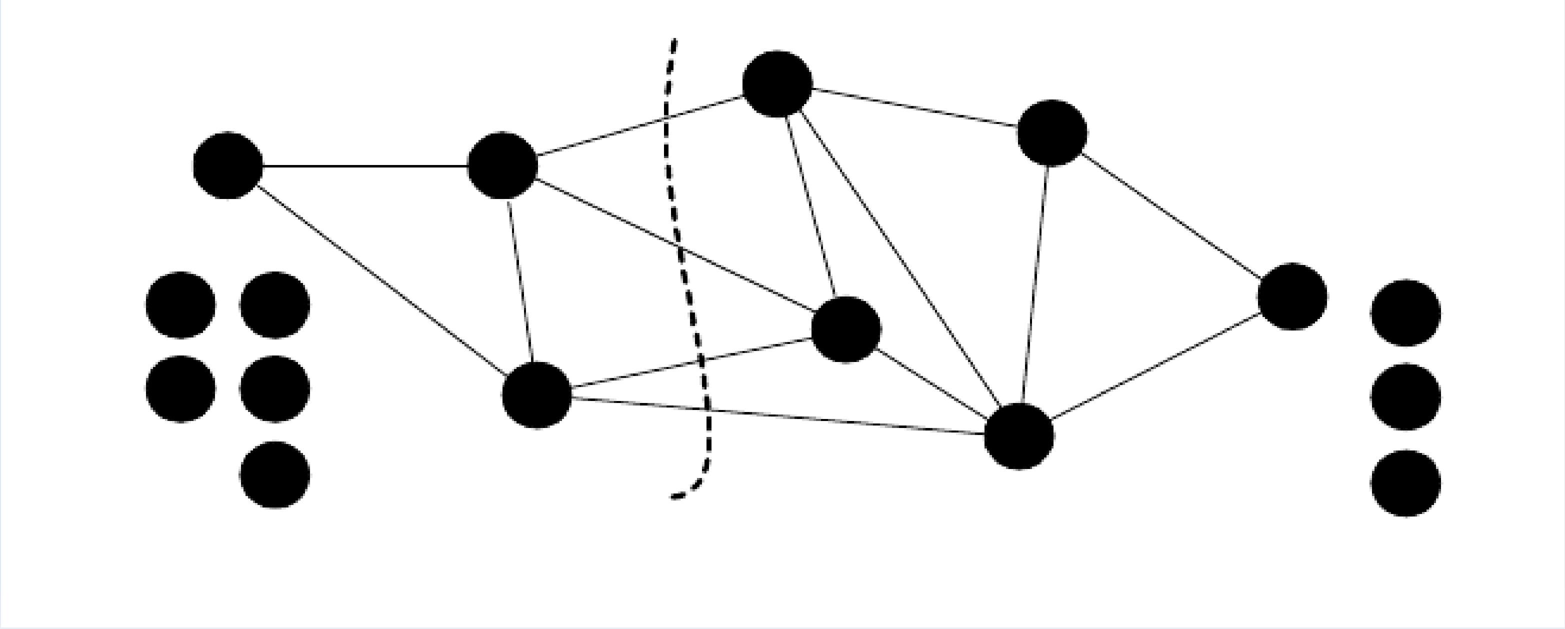
NP-Complete Problems

Max Bisection Is NP-Complete

- max cut becomes max bisection if we require that $|S| = |V - S|$.
- 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)

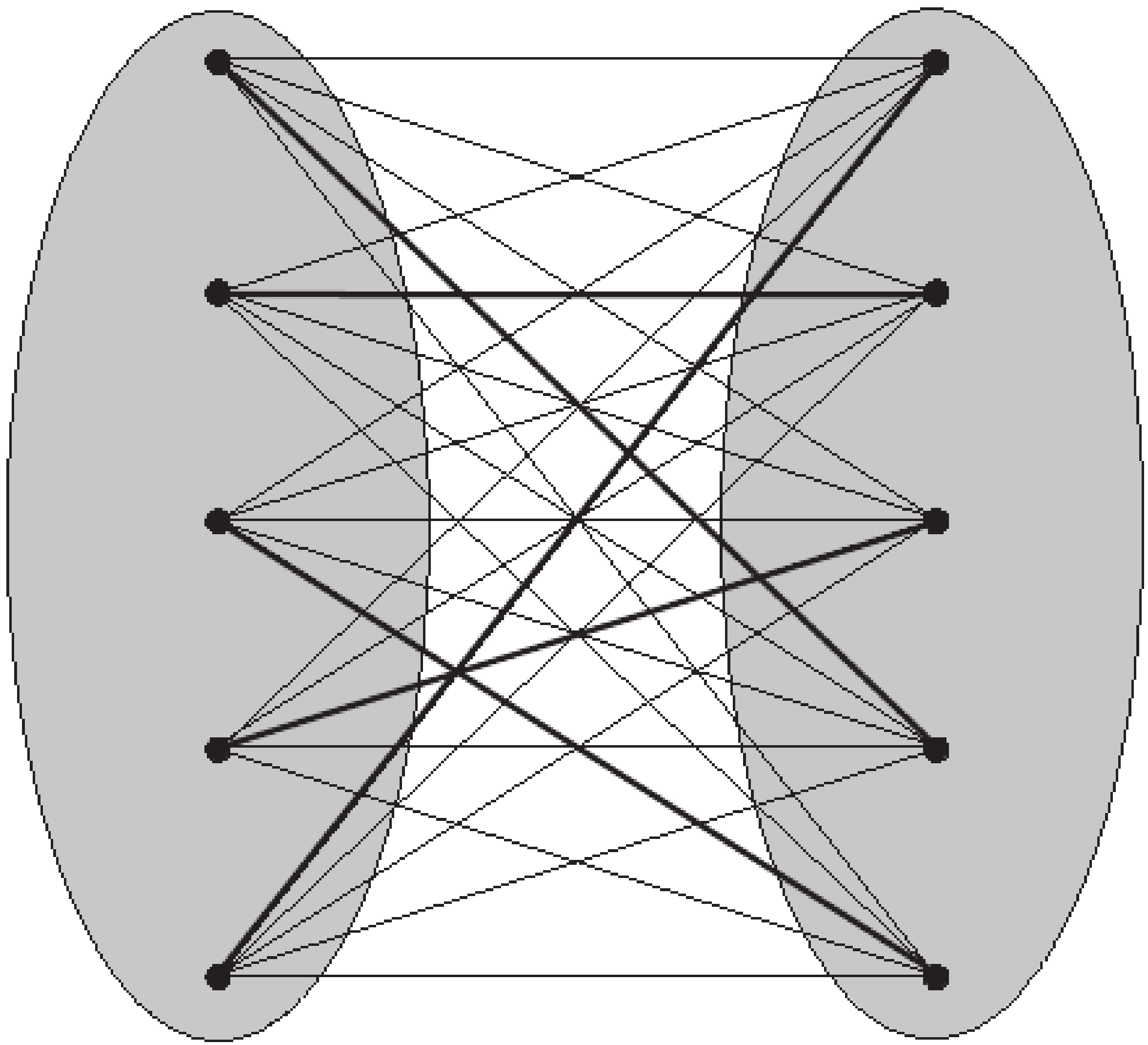
- Every cut (S, V − S) of G = (V, E) can be made into a bisection by appropriately allocating the new nodes between S and V − S.
- Hence each cut of G can be made a cut of G′ of the same size, and vice versa.



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 remainsNP-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 $n^2 - K$.
  - So G has a bisection of size $\geq K$ if and only if its complement has a bisection of size $\leq n^2 - K$.

Illustration

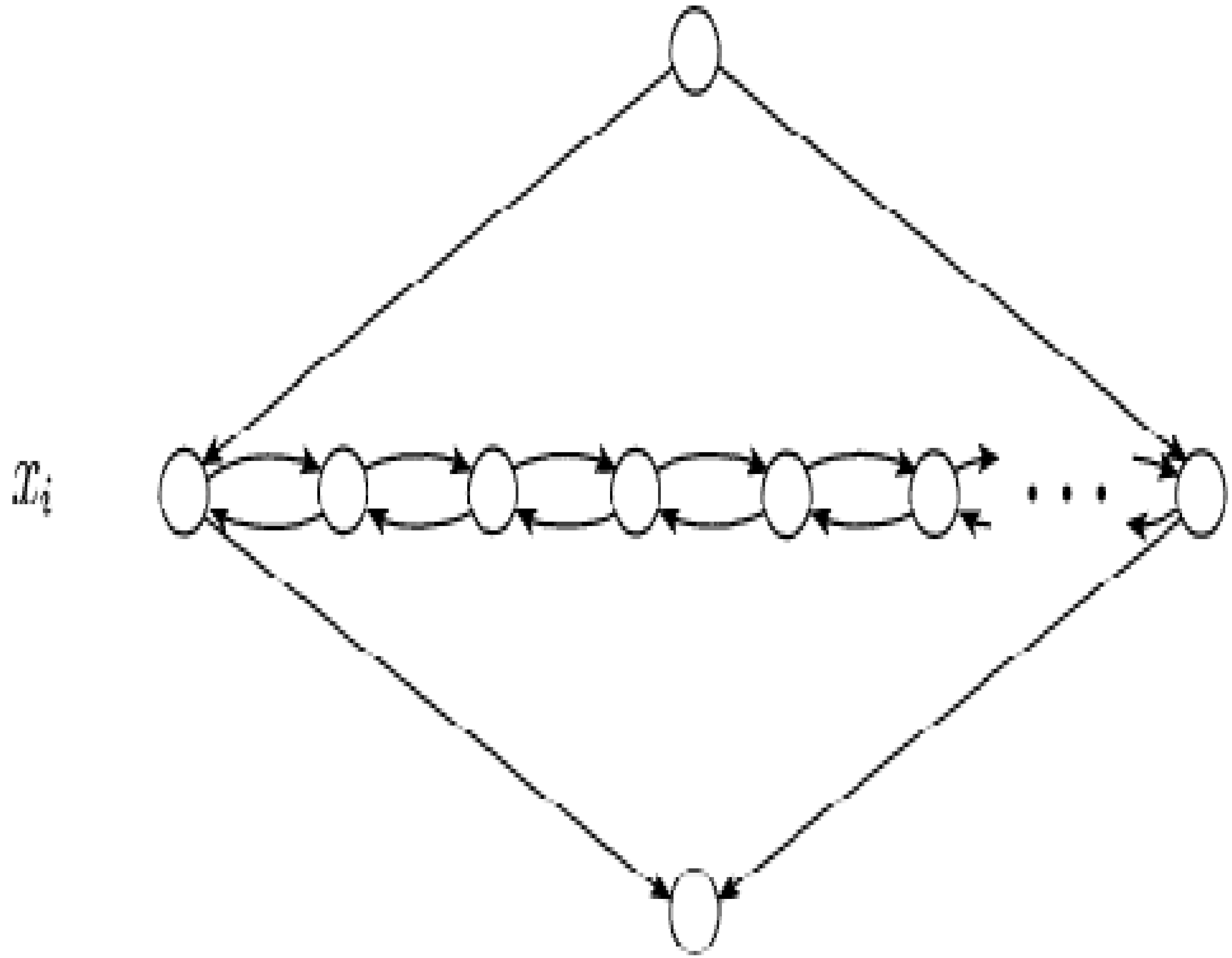


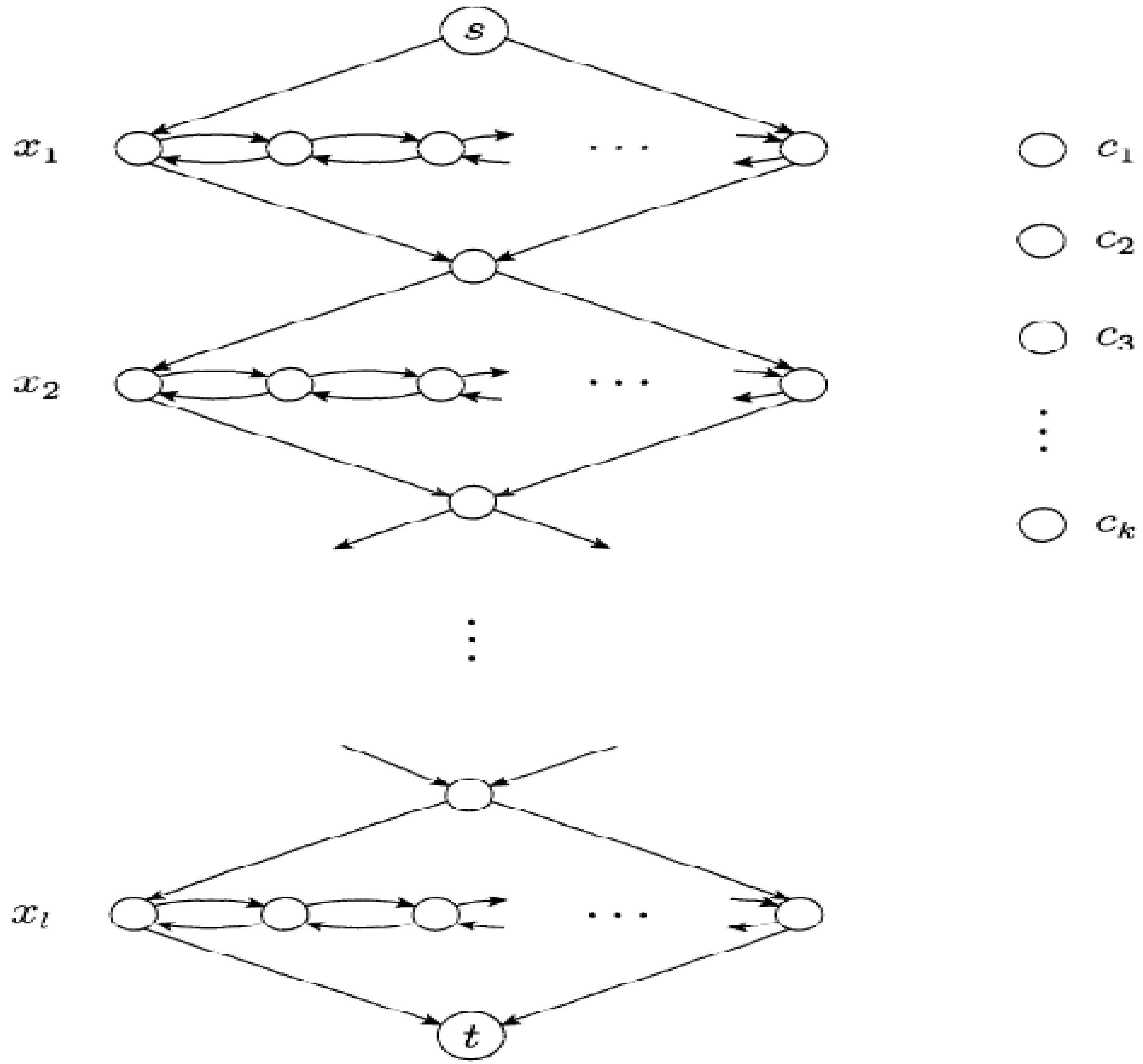
Hamiltonian Path is NP-Complete

- Given an undirected graph, the question whether it has a Hamiltonian path is NP-complete.
  - Karp (1972)
- Hamiltonian Path is in NP (easy)
- Hamiltonian Path is in NP-Hard

Hamiltonian Path is in NP-Hard

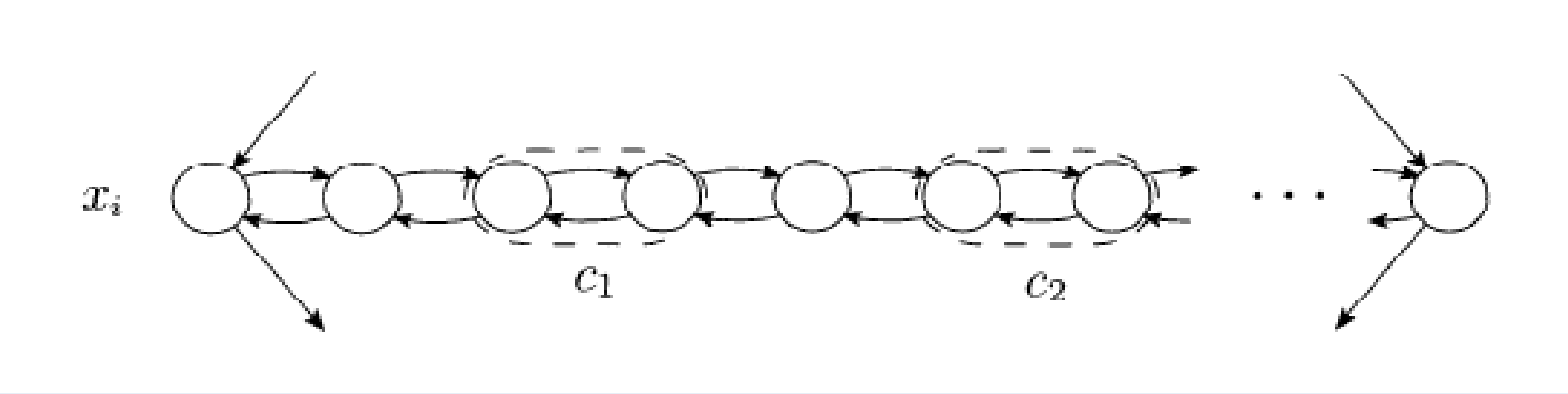
- We reduce 3SAT to HP
- Given a 3-SAT formula φ we construct a directed graph G where a HP exists iff φ is satisfiable.
- φ=(a1 v b1 v c1)...(ak v bk v ck) -contains k clauses
- Each variable xi is represanted with a diamond Di
- Each clause with a single node ci


x_i

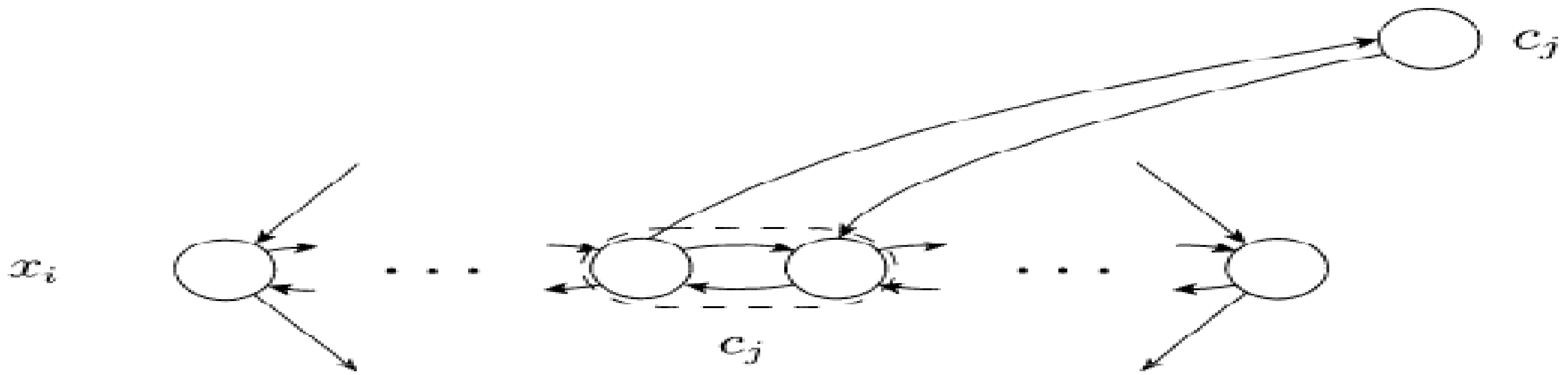

s
x1
x2
xl
t
c1
c2
c3
ck

The chains

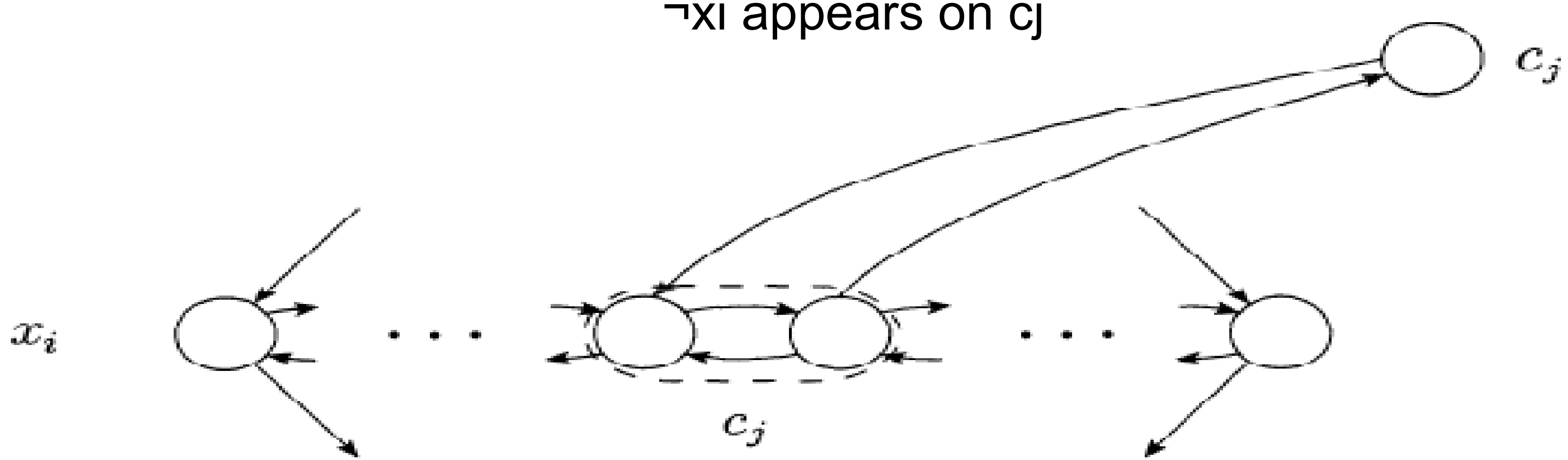
- Each diamond contains a horizontal chain
- The nodes of the chain are grouped in pairs one pair for each clause + separation nodes
- If a variable appears on a clause we connect the clause-pair to the clause-node

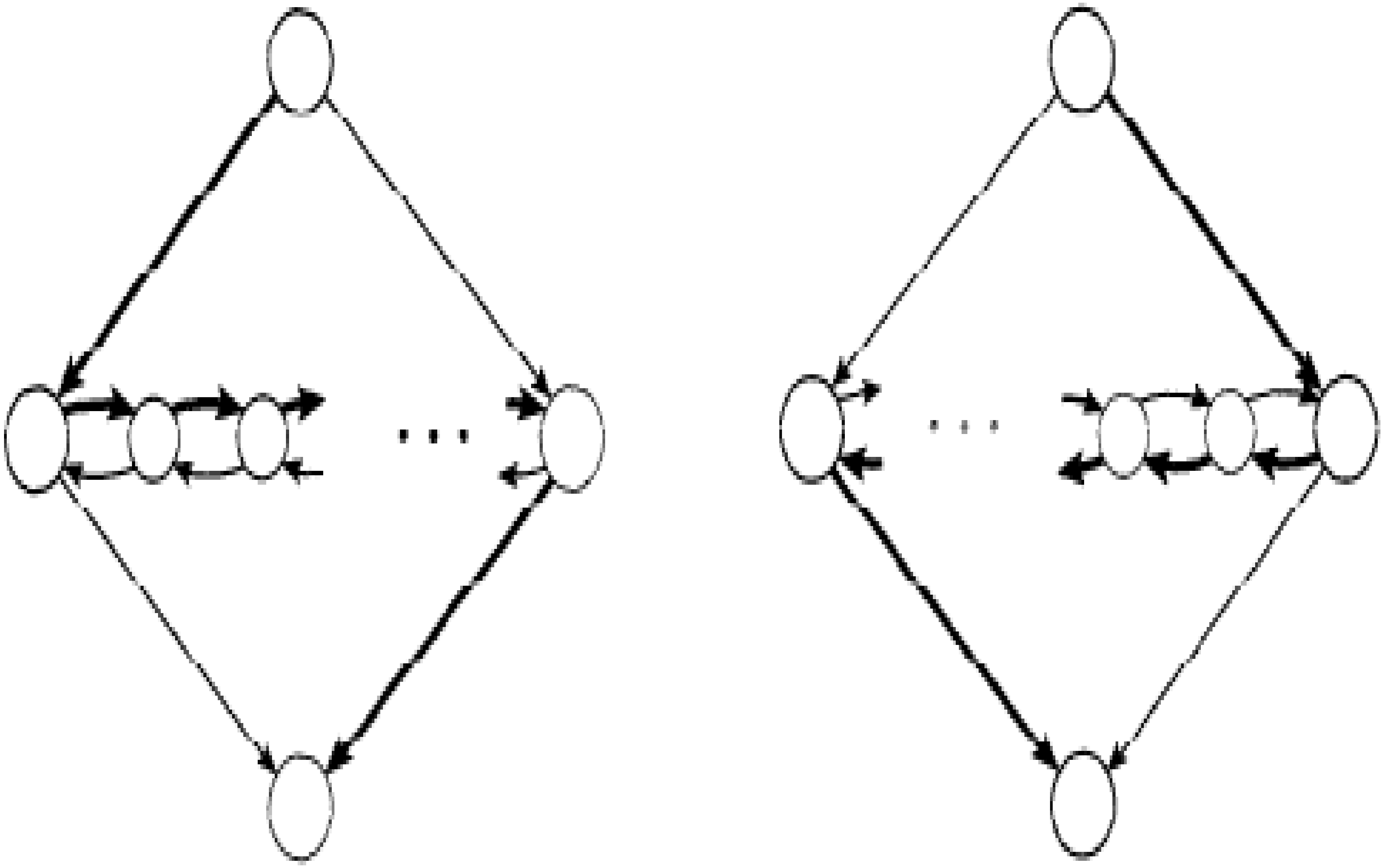


xi appears on Cj



¬xi appears on cj





zig-zag zag-zig

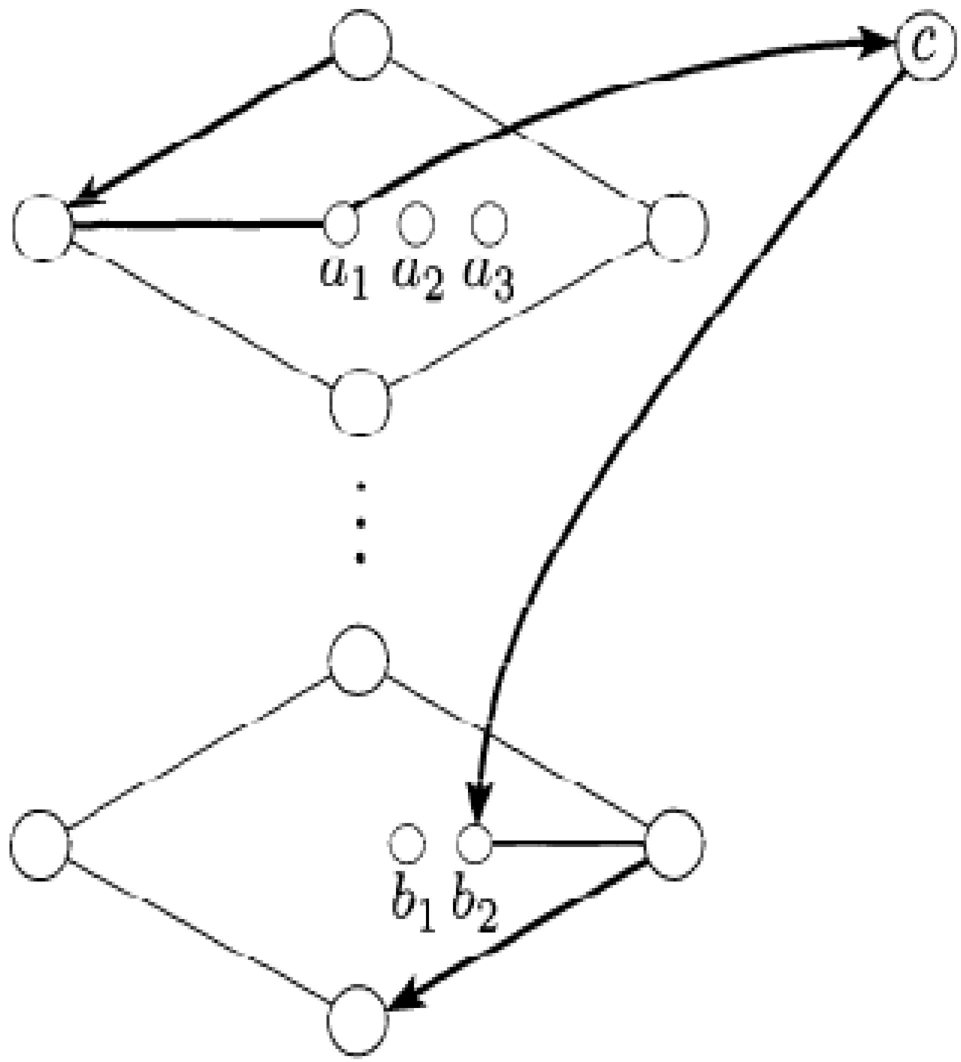
Φ satisfiable => G has a HP

- HP: s->t
- If x is true the path zig-zags otherwise it zag-zigs
- To include the clause nodes we detour on one of the literals that is assigned to be true
- If the literal ¬x is evaluated true we can still detour as we connected with the clause node accordingly

G has an HP => φ Satisfiable

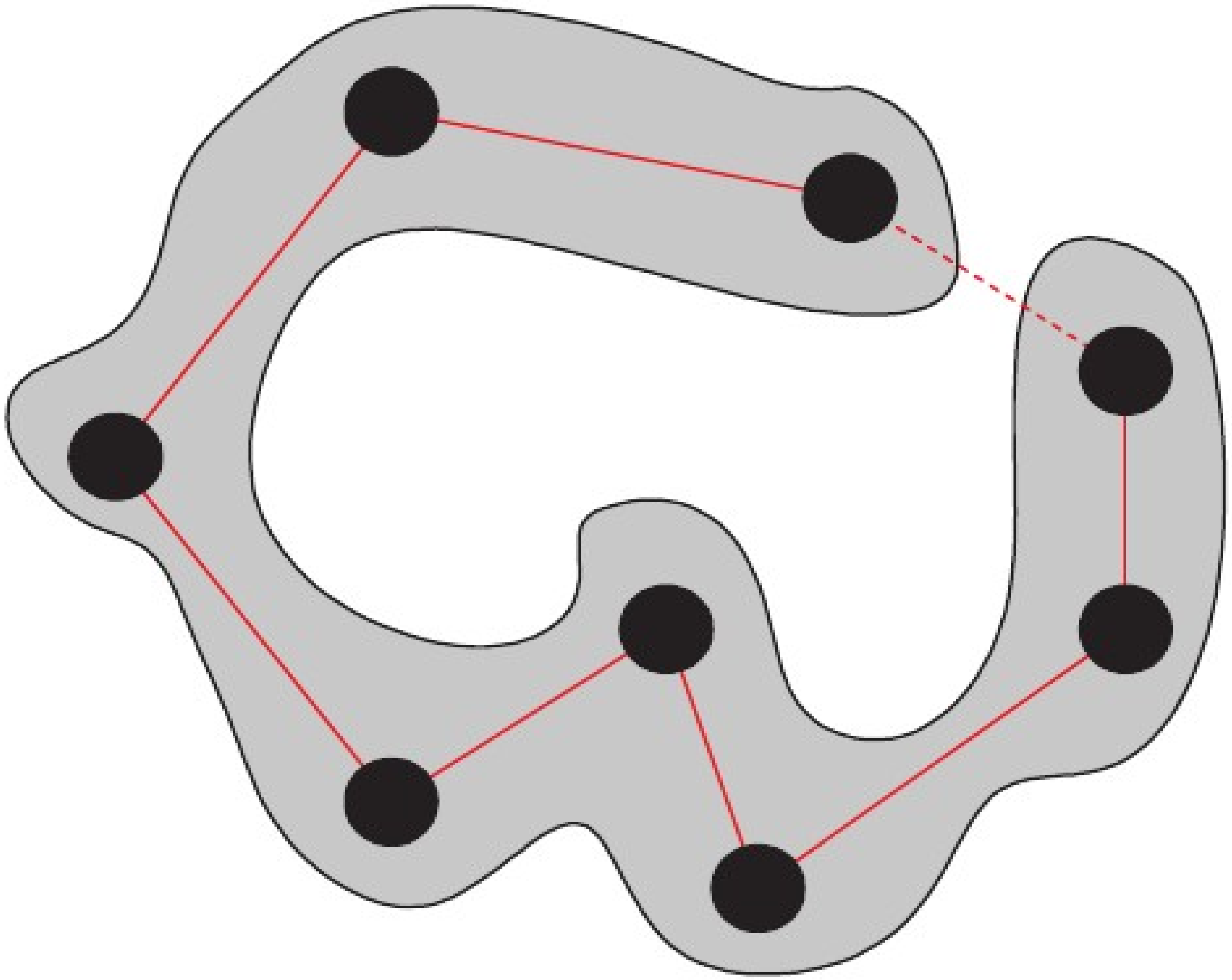
- HP is normal = Traverses the diamonds top to bottom (s->t)
- From a normal HP we obtain the sat assignment
- An HP can only be normal beacause of the seperator nodes

a2 not visitable



TSP (D) Is NP-Complete

- Consider a graph G with n nodes.
- Define dij = 1 if [i, j] ∈ G and dij = 2 if [i, j] \notin G.
- Set the budget B = n + 1.
- Suppose G has no Hamiltonian paths.
- Then every tour on the new graph must contain at least two edges with weight 2.
  - Otherwise, by removing up to one edge with weight 2, one obtains a Hamiltonian path, a contradiction.



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 the new graph 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 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 such that no two adjacent nodes have the same color.
- 2-coloring is in P (find an odd circle).
- But 3-coloring is NP-complete (see next page).
- k-coloring is NP-complete for k ≥ 3.

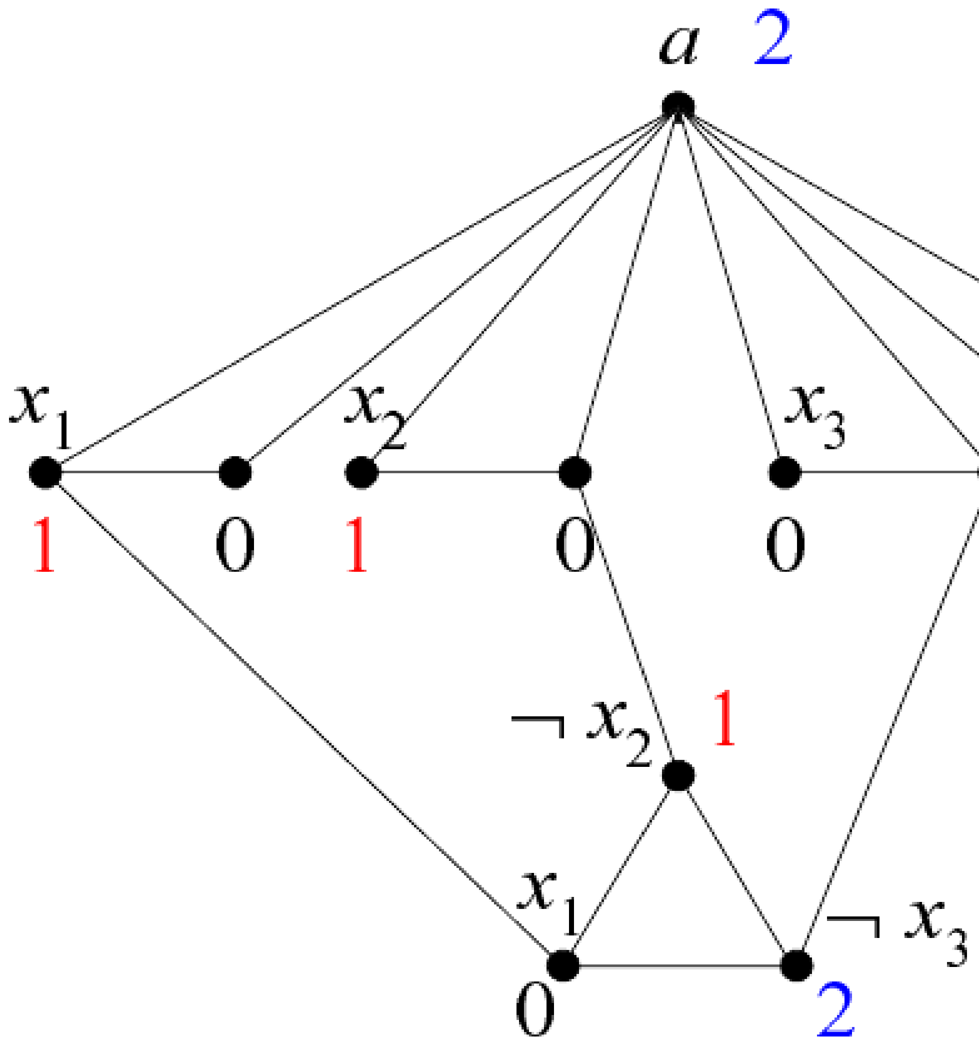
3-coloring Is NP-Complete

- We will reduce NAESAT to 3-coloring.
- We are given a set of clauses C1 , C2 , . . . ,Cm each with 3 literals.
- The boolean variables are x1 , x2 , . . . , xn .
- 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.

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 .

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 xi and ¬xi must take the color 0 and the other 1.

The Proof (continued)

- Treat 1 as true and 0 as false
  - 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.

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 (concluded)

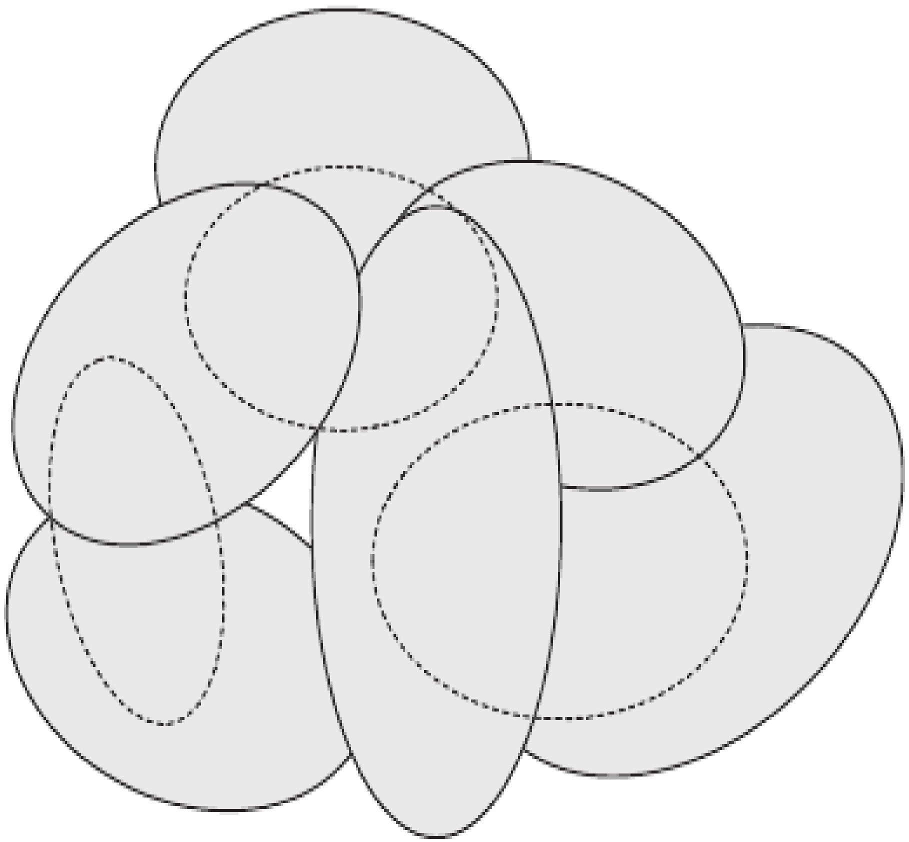
- For each clause triangle:
  - Pick any two literals with opposite truth values.
  - Color the corresponding nodes with 0 if the literal is true and 1 if it is false.
  - Color the remaining node with color 2.
- The coloring is legitimate.
  - If literal w of a clause triangle has color 2, then its 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.

Tripartite Matching

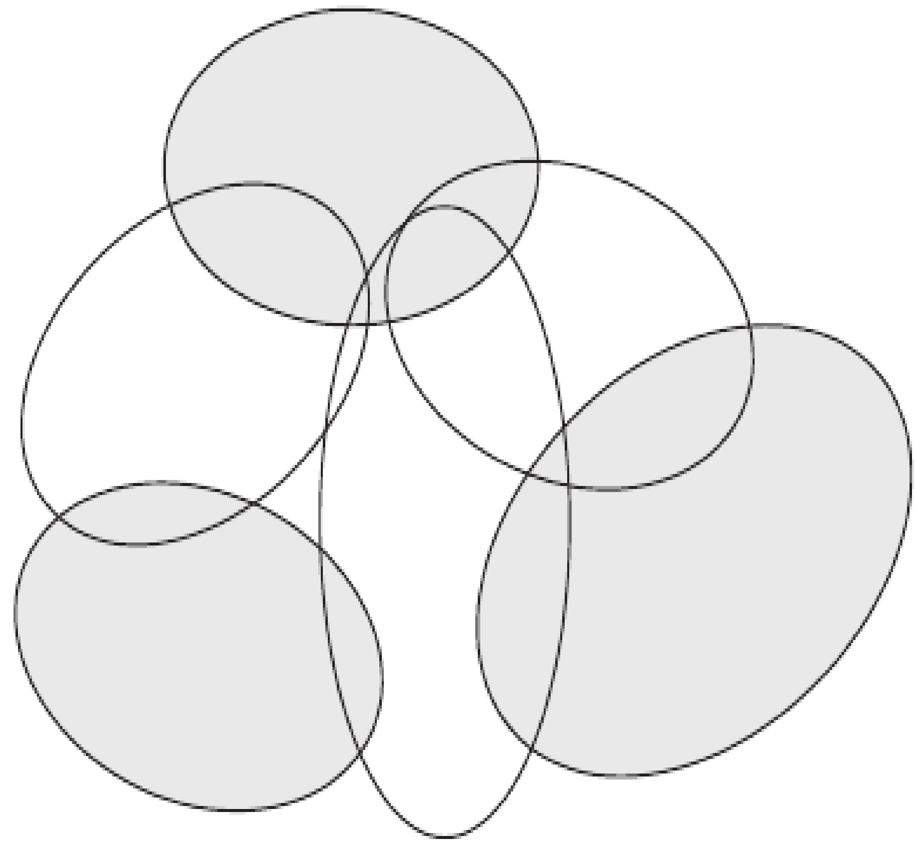
- We are given three sets B, G, and H, each containing n elements.
- Let $T \subseteq B \times G \times H$ be a ternary relation.
- Tripartite Matching asks if there is a set of n triples in T , none of which has a component in common.
  - Each element in B is matched to a different element in G and different element in H.
- Tripartite Matching is NP-complete

Related Problems

- We are given a family $F = \{S1 , S2 , . . . , Sn \}$ 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 $m \in N$ and $|Si | = 3$ 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, and Exact Cover by 3-sets are all NP-complete.



SET COVERING



SET PACKING

The Knapsack Problem

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

Knapsack Is NP-Complete

- Knapsack ∈ NP: Guess an S and verify the constraints.
- We assume $v_i = w_i$ for all i and K = W .
- Knapsack now asks if a subset of $\{v_1 , v_2 , . . . , v_n \}$ adds up to exactly K.
- We shall reduce exact cover by 3-sets to knapsack.

The Proof (continued)

- We are given a family F = {S1 , S2 , . . . , Sn } of size-3 subsets of U = {1, 2, . . . , 3m}.
- Exact Cover by 3-sets asks if there are m disjoint sets in F that cover the set U .
- 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}.
- Our goal is 11 · · · 1.(3m bits)

The Proof (continued)

- 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 occurs when there is carry.
  - 001100010 + 001110000 = 010010010, which denotes the set {2, 5, 8}, not the desired {3, 4, 5, 8}

The Proof (continued)

- Carry may also lead to a situation where we obtain our solution 11 · · · 1 with more than m sets in F .
  - 001100010 + 001110000 + 101100000 + 000001101 = 111111111.
  - But this “solution” {1, 3, 4, 5, 6, 7, 8, 9} does not correspond to an exact cover.
  - And it uses 4 sets instead of the required 3
- To fix this problem, we 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.

The Proof (continued)

- Set v_i to be the (n+1)-ary number corresponding to bit vector encoding Si .
- Now in base n + 1, if there is a set S such that
- $\Sigma_{vi \in S}$ vi = 11 · · · 1, then every bit position must be contributed by exactly one vi and |S| = m.
- Finally, set
- K= $\Sigma_{j=0}^{3m-1}$(n + 1)j = 11 · · · 1 (base n + 1).

The Proof (continued)

- Suppose F admits an exact cover, say {S1 , S2 , . . . , Sm }.
- Then picking S = {v1 , v2 , . . . , vm } clearly results in v1 + v2 + · · · + vm = 11 · · · 1 .
  - It is important to note that the meaning of addition (+) is independent of the base
  - It is just regular addition.

The Proof (concluded)

- On the other hand, suppose there exists an S such that $\Sigma_{v_i \in S} v_i = 11 \cdots 1$ in base $n + 1$.
- The no-carry property implies that $|S| = m$ and $\{S_i : v_i \in S\}$ is an exact cover.

An Example

- Let $m = 3$, $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and
- $S1 = \{1, 3, 4\}$,
- $S2 = \{2, 3, 4\}$,
- $S3 = \{2, 5, 6\}$,
- $S4 = \{6, 7, 8\}$,
- $S5 = \{7, 8, 9\}$.
- Note that $n = 5$, as there are 5 S_i ’s.

An Example (concluded)

- Our reduction produces
- $K = \Sigma 6^j = 11 \cdots 1$ (base 6)
- v1 = 101100000,
- v2 = 011100000,
- v3 = 010011000,
- v4 = 000001110,
- v5 = 000000111.
- Note v1 + v3 + v5 = K.
- Indeed, S1 ∪ S3 ∪ S5 = {1, 2, 3, 4, 5, 6, 7, 8, 9}, an exact cover by 3-sets.