NP-HARD GRAPH PROBLEMS

How to show a problem is NP-hard

- The strategy we adopt to show that a problem L_2 is NP-hard is:
  1. Pick a problem L_1 already known to be NP-hard.
  2. Show how to obtain (in polynomial deterministic time) an instance I' of L_2 from any instance I of L_1 such that from the solution of I' we can determine (in polynomial deterministic time) the solution to instance I of L_1.
  3. Conclude from step (2) that $L_1 \leq L_2$
  4. Conclude from steps (1) and (3) and the transitivity of $\leq$ that L_2 is NP-hard.

The strategy to show that a problem L_2 is NP-hard is

(i) Pick a problem L_1 already known to be NP-hard.

(ii) Show how to obtain an instance I^1 of L_2 from any instance I of L_1 such that from the solution of I^1

- We can determine (in polynomial deterministic time) the solution to instance I of L_1.

(iii) Conclude from (ii) that $L_1 \alpha L_2$.

(iv) Conclude from (i),(ii), and the transitivity of α that satisfiability αL_1

$L_1 \alpha L_2$

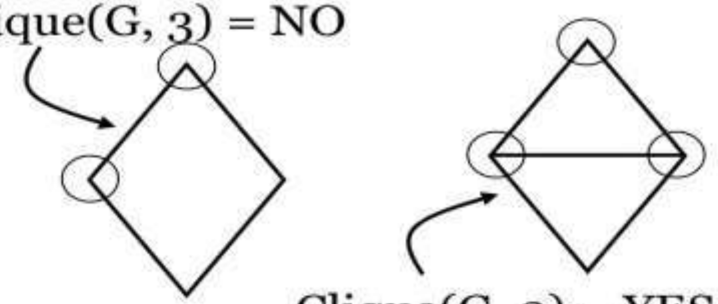
$\therefore$ Satisfiability αL_2

$\therefore$ L_2 is NP-hard

Clique Decision Problem(CDP)

- Clique Problem:
  - Undirected graph G = (V, E)
  - Clique: a subset of vertices in V all connected to each other by edges in E (i.e., forming a complete graph)
  - Size of a clique: number of vertices it contains
- Optimization problem:
  - Find a clique of maximum size
- Decision problem:
  - Does G have a clique of size k?

Clique(G, 2) = YES
Clique(G, 3) = NO



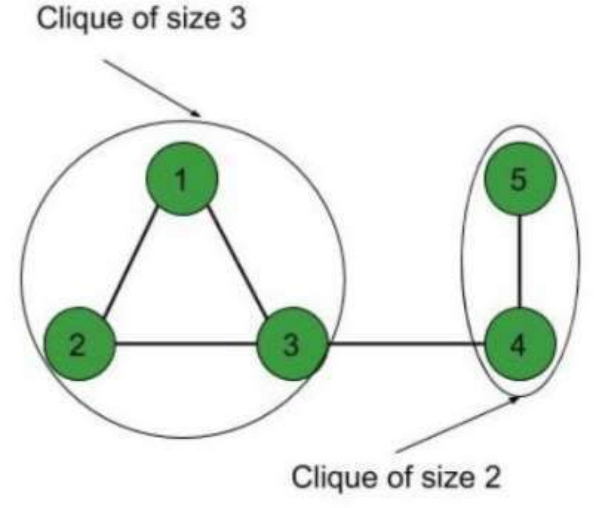
Clique(G, 3) = YES
Clique(G, 4) = NO

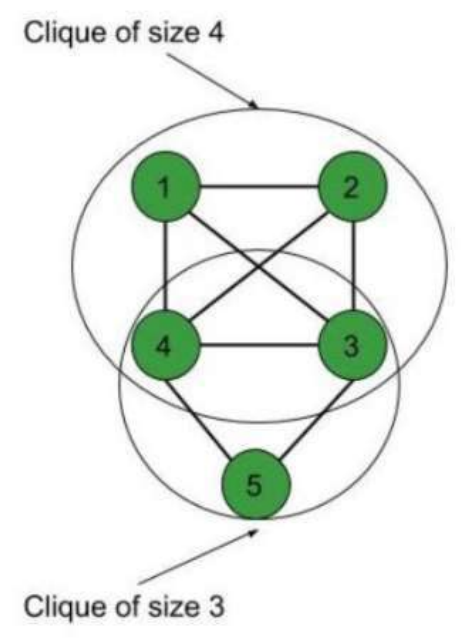
Clique Decision Problem(CDP) :

Clique:

Clique is a maximal complete sub graph of a graph G = (V,E)

– Size of a clique is the number of vertices in it





KL DEEMED TO BE UNIVERSITY

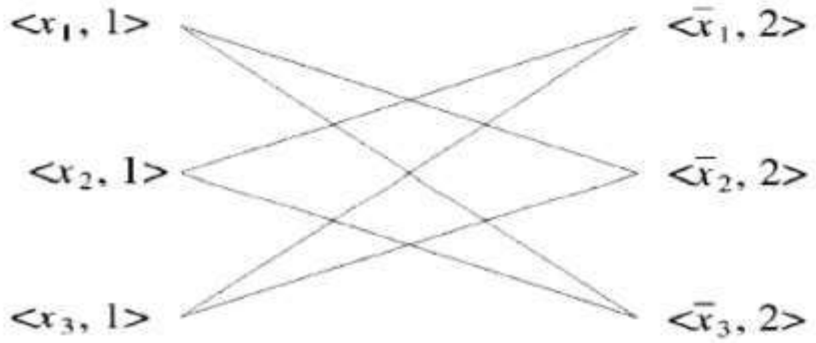
Clique Decision Problem(CDP)

- **Theorem:** CNF-satidfiability ≤ CDP
- **Proof:** Let $F = \wedge_{1 \leq i \leq k} C_i$ be a propositional formula in CNF. Let x_i, $1 \leq i \leq n$, be the variables in F.
  - We show how to construct from F a graph G = (V, E) such that G has a clique of size at least k if and only if F is satisfiable.
  - If the length of F is m, then G is obtainable from F in O(m) time.
  - Hence, if we have a polynomial time algorithm for CDP, then we can obtain a polynomial time algorithm for CNF-satisfiability using this construction.

Clique Decision Problem(CDP)

- For any F, G = {V,E) is defined as follows:
  - V = {<σ, i> | σ is a literal in clause C_i}
  - E = {(<σ,i>, <δ,j>) | i≠j and σ≠δ(bar)}.

$$F = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$



Example:

Consider $F = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3})$.

The construction of Theorem yields the graph:

This graph contains six cliques of size two.

Consider the clique with vertices $\{\langle x_1, 1\rangle, \langle \overline{x_2}, 2\rangle\}$.

By setting x_1 = true and $\overline{x_2}$ = true (i.e. x_2 = false)

F is satisfied.

x_3 may be set either to true or false.

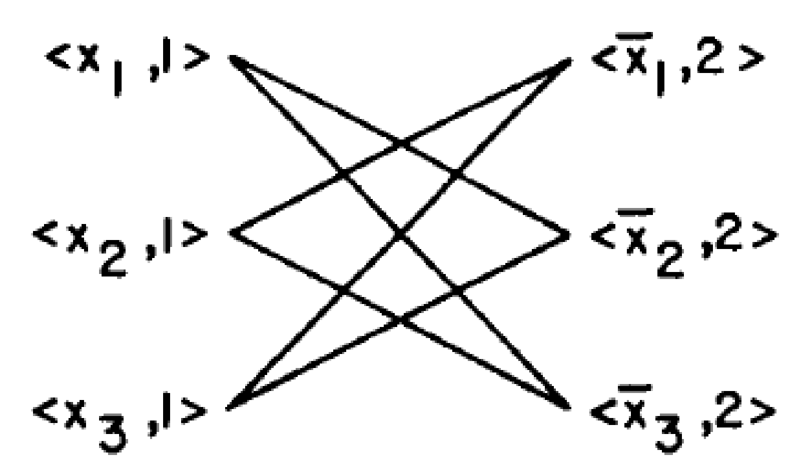


Figure: A sample graph and satisfiability

Theorem: CNF- satisfiability α Clique Decision Problem

Proof:

Let $F = \wedge_{1 \le i \le k} C_i$ be a propositional formula in CNF. Let x_i, $1 \le i \le n$ be the variables in F.

We shall show how to construct from F a graph G = (V, E) such that G will have a clique of size at least k if F is satisfiable.

If the length of F is m, then G will be obtainable from F in O(m) time.

Hence, if we have a polynomial time algorithm for CDP, then we can obtain a polynomial time algorithm for CNF-satisfiability using this construction.

For any F, G = (V, E) is defined as follows: $V = \{\langle \sigma, i\rangle | \sigma$ is a literal in clause $C_i\}$; $E = \{(\langle \sigma, i\rangle, \langle \delta, j\rangle) | i \neq j$ and $\sigma \neq \delta\}$.

A sample construction is given in Example.

Questions:

1. Discuss in detail about Clique Decision Problem
2. Reduce CNF-Satisfiability problem into CDP and Solve

THANK YOU