

NP - Completeness

Independent Set

Input: • Graph $G = (V, E)$ with n vertices, m edges
• integer $1 \leq k \leq n$

Solution: Independent set of size k (k vertices in V w/ no edges between them)

Algorithm: Try all sets of k vertices

$$\# \text{ of sets} = \binom{n}{k} \approx n^k \Rightarrow \text{runs in time } \Omega(n^k)$$

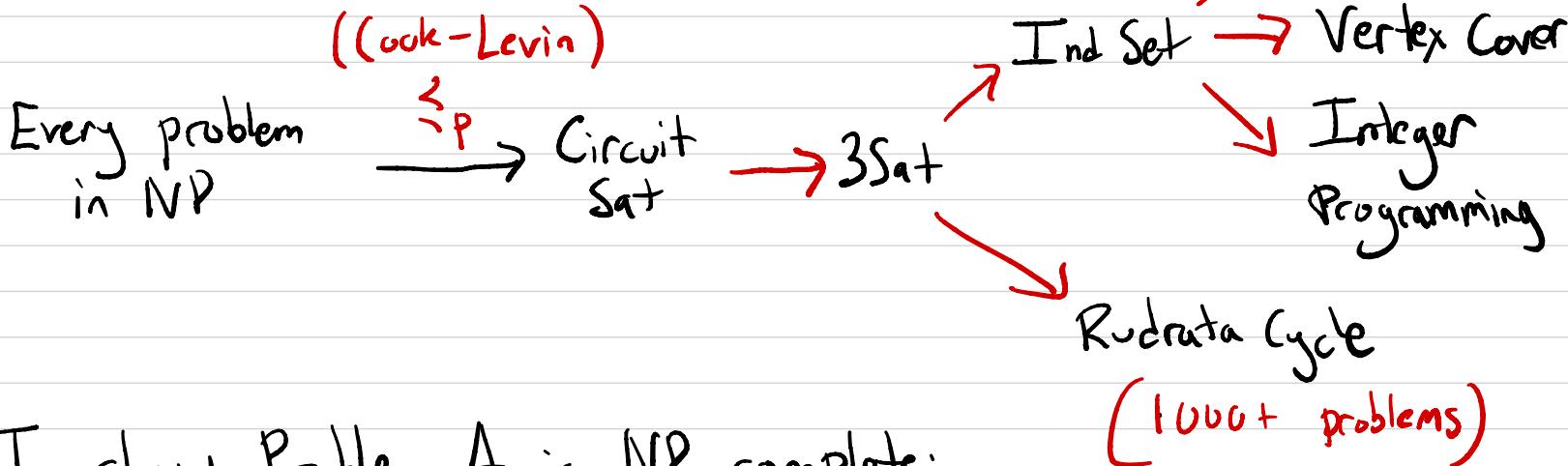
(good if $k = O(1)$, bad if $k = \omega(1)$)

Seems hard! How to show this?

Thm: Independent Set is NP-hard (+ NP-complete)

Pf: Pick some (well-known) NP-complete problem A
and show that $A \leq_p$ Independent Set

NP-completeness



To show Problem A is NP-complete:

1.) $A \in \text{NP}$ (show verification algorithm)

2.) Show NP complete B reduces to A ($B \leq_p A$)

Shows poly-time alg for A \Rightarrow poly-time alg for all of NP
(unlikely!)

REDUCIBILITY AMONG COMBINATORIAL PROBLEMS[†]

Richard M. Karp

University of California at Berkeley

1972

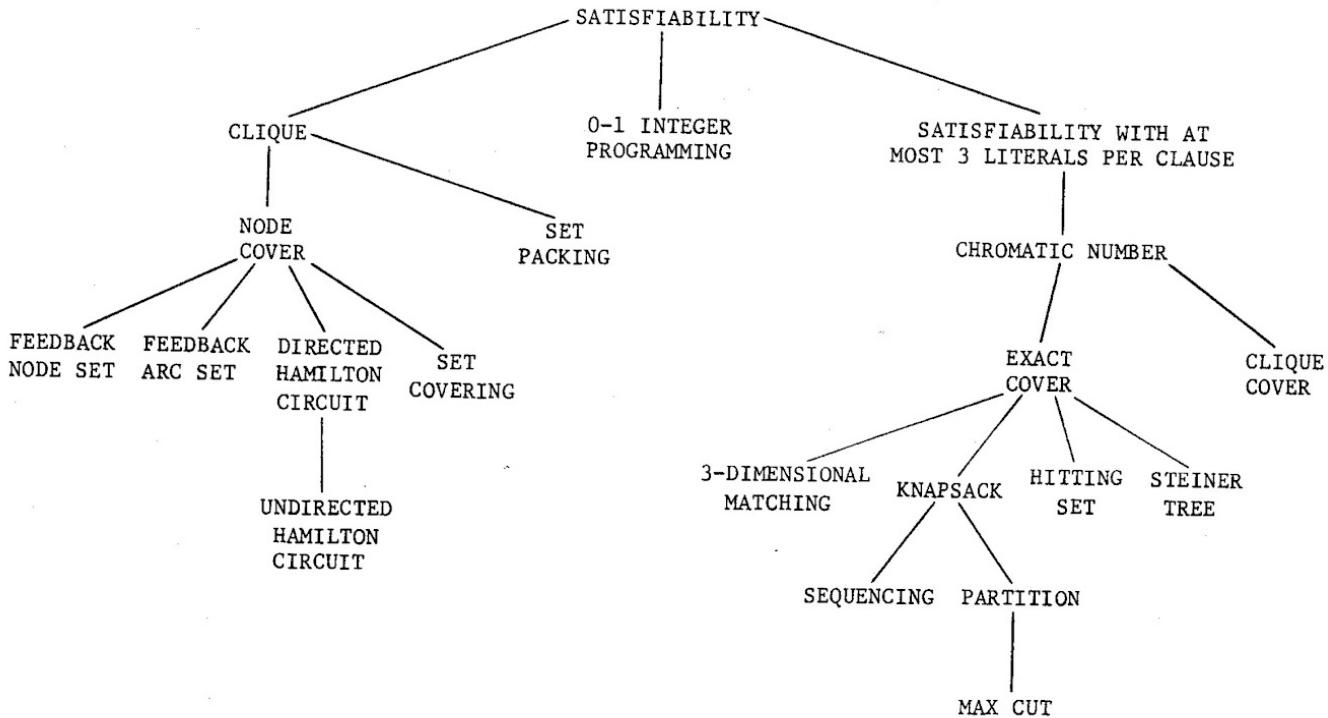


FIGURE 1 - Complete Problems

Thm: 3Sat \leq_p Ind-Set (\because Ind-Set is NP-complete)

3Sat instance +

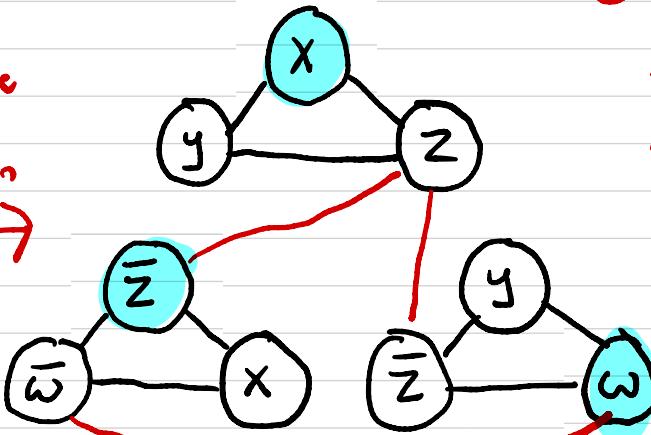
poly-time

Reduction

$$\begin{array}{c} \begin{array}{ccc} & 1 & 0 \\ (x & \vee & y \vee z) \\ | & | & | \\ 1 & 1 & 1 \\ \wedge (\bar{z} & \vee & \bar{w} \vee x) \\ | & | & 0 \\ 1 & 1 & \\ \wedge (y & \vee & \bar{z} \vee w) \end{array} \end{array}$$

Sat assign: $x=y=1, z=w=0$

Ind Set instance G



poly-time

Recovery

$$x = 1$$

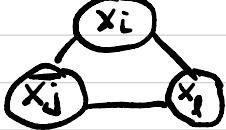
$$y = 0 \text{ or } 1$$

$$z = 0$$

$$w = 1$$

$K = m = \# \text{ of clauses}$

Proof part 1 : If 3Sat inst x has sat assign $A: \{x_1, \dots, x_m\} \rightarrow \{0, 1\}$, then graph G has ind set I of size $k=m$

- For each clause $x_i \vee \bar{x}_j \vee x_k$, A sets either $x_i=1, \bar{x}_j=1, x_k=1$.
- Pick one (say x_i). Add $\circlearrowleft x_i$ of  to I .
- So I is of size $k=m$.
- I contains no $\circlearrowleft x_i - \circlearrowleft \bar{x}_i$. \therefore No edges, so independent set.

Proof part 2: Let I be ind set in G of size K .

Then Recovery alg outputs satisfying assignment.

Recovery alg

$$\text{variable } x_L = \begin{cases} 1 & \text{if some } \circlearrowleft x_i \in I \\ 0 & \text{if some } \circlearrowleft \bar{x}_i \in I \\ \text{arbitrary o.w.} & \end{cases}$$

- For each i , only $\circlearrowleft x_i$ or $\circlearrowleft \bar{x}_i \in I$. So Recovery alg well-defined.

- For each , $x_i=1, x_j=1$, or $x_k=1$. So all clauses satisfied.

Integer Programming (IP)

Input: A linear program

Solution: An integer solution to LP

Thm: Ind-Set \leq_p Integer Programming

Pf: Ind Set instance

- $G = (V, E)$
- $V = \{1, 2, \dots, n\}$
- integer k

poly-time
Reduction

IP instance

$$0 \leq x_i \leq 1$$
$$\sum_{i=1}^n x_i = k$$

$$\forall (i, j) \in E, x_i + x_j \leq 1$$

Proof part 1: If \exists ind set I of size k ,
 \exists solution to IP instance. (Just set $x_i = \begin{cases} 1 & \text{if } i \in I \\ 0 & \text{if } i \notin I \end{cases}$)

Proof part 2: If (x_1, \dots, x_n) is solution to IP,
can "recover" and ind set of size k . (Just put $i \in I$ if $x_i = 1$)

Clique

Input:

- graph $G = (V, E)$
- integer $1 \leq k \leq n$

(k vertices in V in which every pair is adjacent)

Output: Clique of size k



= 5-clique

Thm: Independent Set \leq_p Clique

Pf: Ind set instance

- $G = (V, E)$
- integer k

Reduction alg

Clique instance

- $\bar{G} = (V, \bar{E})$
- \bar{E} = set of edges not in E
- same integer k

Ind set in G : $\leftarrow \rightarrow$ Clique in \bar{G} Δ

Recovery alg: Given clique $S \subseteq V$, output S .

Rudrata (s,t) path

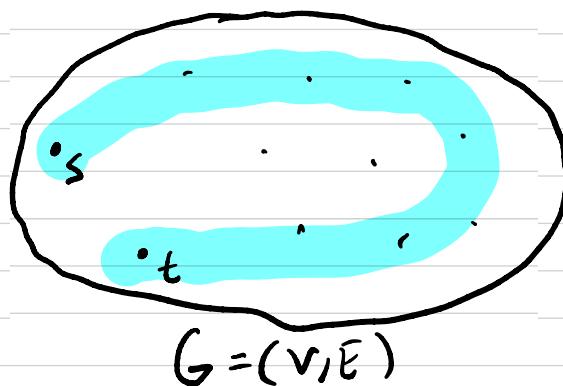
Input: Graph $G=(V,E)$, source $s \in V$, destination $t \in V$

Output: path starting at s , ending at t , visiting each vertex exactly once

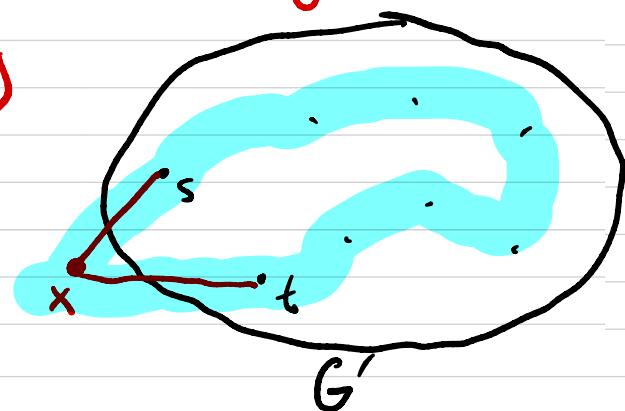
Thm: Rudrata (s,t) path \leq_p Rudrata Cycle

Pf: Rud (s,t) Path instance

Rud cycle instance



Reduction alg
→

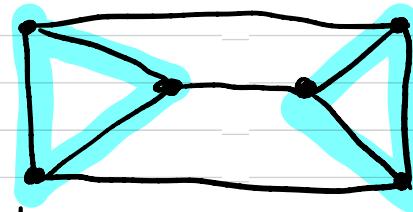


Recovery alg: given cycle in G' , delete (x,s) and (x,t) edges

Double Cycle

Input: Graph $G=(V,E)$ with $n=|V|=\text{even}$

Output: two disjoint cycles of $n/2$ vertices each,
which visit each vertex once



Thm: Rudrata Cycle \leq_p Double Cycle

Pf: Rudrata instance

Reduction alg

Double Cycle Instance

