F, 11/15/19

Fall'19 CSCE 629

Analysis of Algorithms

Fang Song
Texas A&M U

Lecture 30

- Hamiltonian Cycle
- Approximating vertex cover

Quiz

Suppose P ≠ NP. Which of the following are still possible?

a) $O(n^3)$ algorithm for factoring n-bit integers.
b) $O(1.657^n)$ time algorithm for $\mathrm{HAM\text{-}CYCLE}$.
c) $O(n^{\log \log \log n})$ algorithm for 3-SAT.
d) There exist problems that are neither in $\mathbf{P}$ nor $\mathbf{NP}$.

https://en.wikipedia.org/wiki/NP-intermediate

Recall: $\mathbf{3-SAT}$ is NP-Complete

Theorem. $\mathbf{3-SAT}$ is NP-Complete

Pf. We show $\text{Circuit}-\text{SAT} \leq_P 3-\text{SAT}$

- Given a circuit K, create a $3-\text{SAT}$ variable x_i for each gate
- Make circuit compute correct values at each node

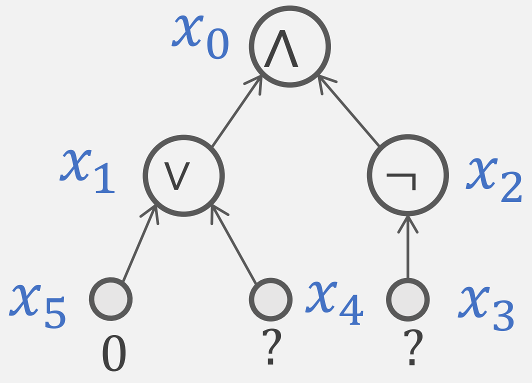
$x_2 = \neg x_3$	$\Rightarrow x_2 \vee x_3, \overline{x_2} \vee \overline{x_3}$
$x_1 = x_4 \vee x_5$	$\Rightarrow x_1 \vee \overline{x_4}, x_1 \vee \overline{x_5}, \overline{x_1} \vee x_4 \vee x_5$
$x_0 = x_1 \wedge x_2$	$\Rightarrow \overline{x_0} \vee x_1, \overline{x_0} \vee x_2, x_0 \vee \overline{x_1} \vee \overline{x_2}$

- Hard-coded input values and output value

$x_5 = 0 \Rightarrow \overline{x_5}$ $\qquad x_0 = 1 \Rightarrow x_0$

- Final step: turn clauses into exactly 3 literals by adding dummy variables

EX. $x_1 \vee x_2 \Rightarrow x_1 \vee x_2 \vee y, x_1 \vee x_2 \vee \overline{y}$



Circuit K satisfiable iff. $\exists$ truth assignment satisfying all clauses constructed

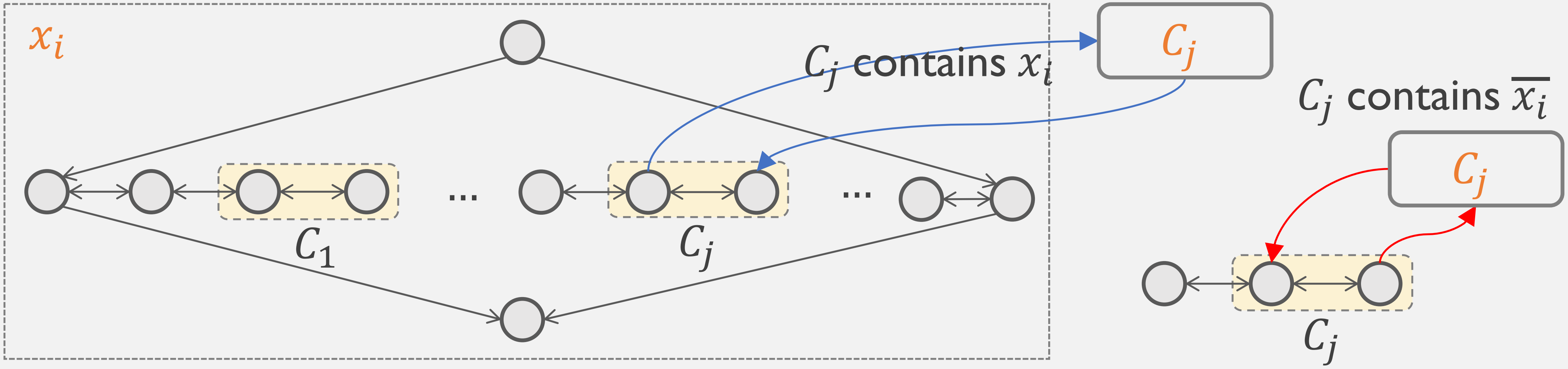
! Don't forget to show $\mathbf{3-SAT} \in \mathbf{NP}$

(DIR−)HAM−CYCLE is NP-Complete

(DIR−)HAM−CYCLE. Given a directed graph $G = (V, E)$, does there exist a directed cycle Γ that visits every node exactly once?

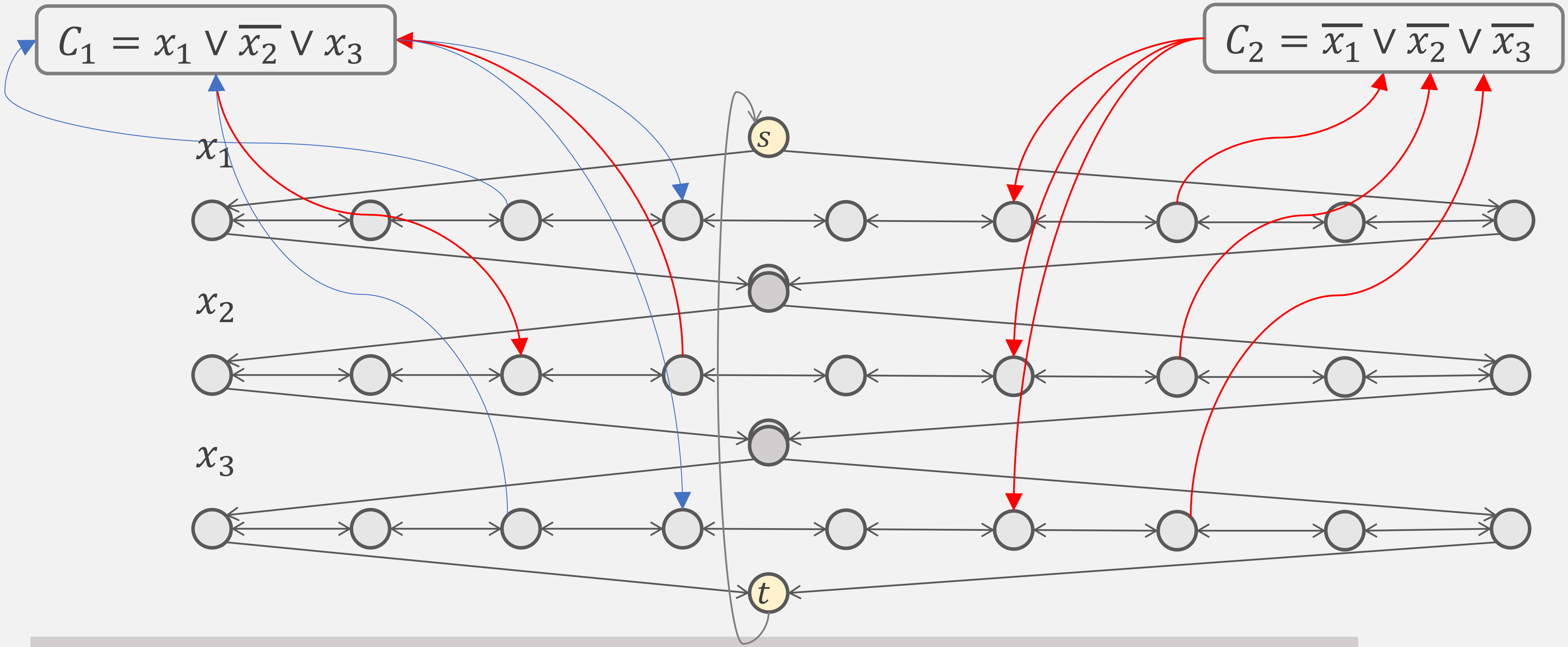
Theorem. $3\text{−SAT} \leq_P \text{(DIR−)HAM−CYCLE}$

Pf. Given 3−SAT instance Φ in CNF: n variables x_i and k clauses C_j



Intuition: traverse row i from left to right $\Leftrightarrow$ set variable $x_i = \text{true}$

$3\text{−SAT} \le_P (\text{DIR−})\text{HAM−CYCLE}$



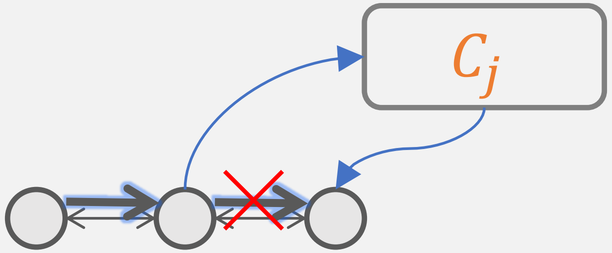
Claim. Φ is satisfiable iff. G has a Hamiltonian cycle

$3-\mathrm{SAT} \le_P (\mathrm{DIR}-)\mathrm{HAM}-\mathrm{CYCLE}$

Claim. Φ is satisfiable iff. G has a Hamiltonian cycle

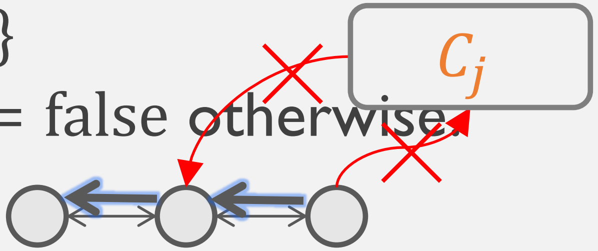
(⇒) Suppose Φ has a satisfying assign. $\boldsymbol{x}^*$. Define an H-Cycle in G:

- if $x_i^* = \text{true}$, traverse row x_i from left to right
- if $x_i^* = \text{false}$, traverse row x_i from right to left
- For each clause C_j pick (only) one row i and take a detour



(⇐) Suppose G has a H-Cycle Γ. Define a satisfying assign. in Φ:

- In Γ, replace edges going/leaving C_j with the edge of the corresponding two nodes in some row. This gives a new cycle Γ' in $G - \{C_1, C_2, \dots, C_k\}$
- In Γ', set $x_i = \text{true}$ if Γ' traverses row i left-to-right; set $x_i = \text{false}$ otherwise.

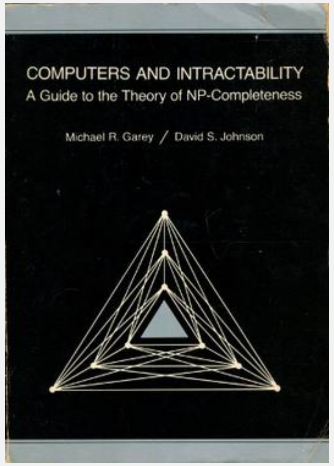


More hard computational problems

- Aerospace engineering: optimal mesh partitioning for finite elements.
- Chemical engineering: heat exchanger network synthesis
- Civil engineering: equilibrium of urban traffic flow [very much needed in BCS!]
- Electrical engineering: VLSI layout.
- Mechanical engineering: structure of turbulence in sheared flows
- Biology: protein folding
- Physics: partition function of 3-D Ising model in statistical mechanics.
- Economics: computation of arbitrage in financial markets with friction
- Financial engineering: find minimum risk portfolio of given return
- Politics: Shapley-Shubik voting power
- Pop culture: Sudoku (http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/SIGAL87-2.pdf)

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Want to learn more?

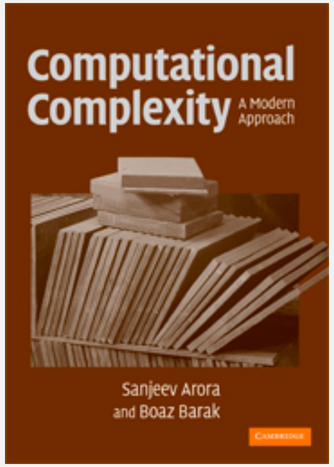


Computers and Intractability: A Guide to the Theory of NP-Completeness.
Michael Garey and David S. Johnson

Most Cited Computer Science Citations

This list is generated from documents in the CiteSeer^{x} database as of March 19, 2015. This list is automaticall
mode and citation counts may differ from those currently in the CiteSeer^{x} database, since the database is con
All Years | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 20
| 2015

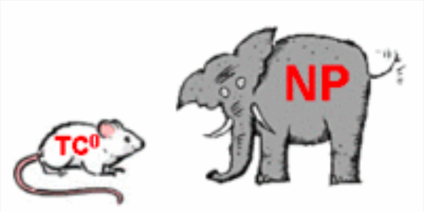
1. M R Garey, D S Johnson
Computers and Intractability: A Guide to the Theory of NPCompleteness" W.H. Feeman and 1979
11468



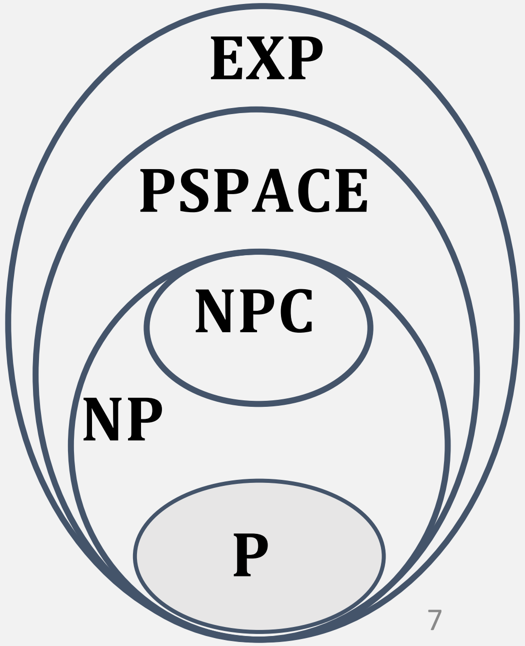
Computational Complexity: A Modern Approach
Sanjeev Arora & Boaz Barak

Spring'20 @TAMU

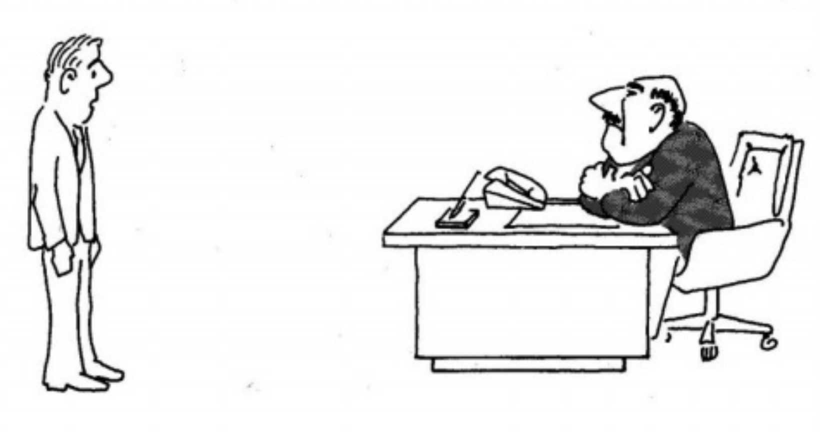
- CSCE 627 Theory of Computability
- CSCE 637 Complexity Theory



Complexity Zoo
There are now 544 classes and counting!



Coping with NP-Completeness



"I can't find an efficient algorithm, I guess I'm just too dumb."

"I can't find an efficient algorithm, but neither can all these famous people."

https://i.stack.imgur.com/EkpIV.jpg

- Better (constructive) answers: sacrifice one of three desired features
  1. ~~Solve arbitrary instances~~
  2. ~~Solve problems in poly-time~~
  3. ~~Solve problems to optimality~~

- Techniques
  - Identifying structured special cases
  - Local search heuristics (e.g., gradient descent)
  - Approximation algorithms

Finding near-optimal vertex cover

Input. Graph $G = (V, E)$ and an integer k

- Vertex cover $S \subseteq V$: subset of vertices that touches all edges

Goal. Find a vertex cover S of minimum size

- First attempt: greedily pick the vertex that covers most edges

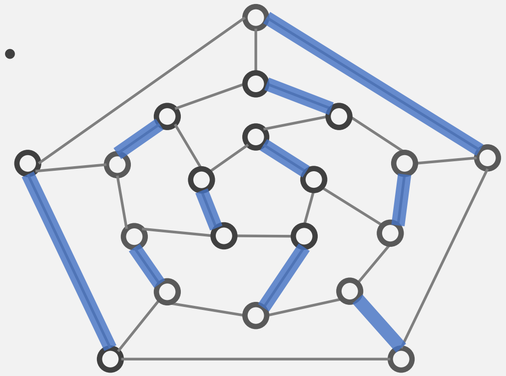
> APP-VC: on input $G = (V, E)$
>   For $v \in V$ (in descending order of degrees)
>   Add v in S
>   Delete v and its neighbors from G

- Claim. Suppose the minimum vertex cover has size OPT. Then the output of APP-VC has size at most $O(\log n \cdot \mathrm{OPT})$
- Pf. Exercise (Hint on board)

2-approximation vertex cover

Recall: $M \subseteq E$ is a matching in $G = (V, E)$ if each node appears in at most one edge in M.

Observation: For any matching M and any vertex cover S, $|M| \leq |S|$. In particular, $|M| \leq \mathrm{OPT}$ (size of min vertex cover).



- 2nd attempt: find a MAX matching

> 2-APP-VC: on input $G = (V, E)$
> Find a maximal matching $M \subseteq E$
> Return $S = \{\text{all end points of edges in } M\}$

- Claim. The output of 2-APP-VC has size at most $2 \cdot \mathrm{OPT}$
- Pf. $|S| = 2|M| \leq 2 \cdot \mathrm{OPT}$. Why does S have to be a vertex cover?
  - Exercise. Is this tight, i.e., 2-APP-VC's output $= 2 \cdot \mathrm{OPT}$ on some graph?