CSE 6140/ CX 4140:

Computational Science and Engineering ALGORITHMS

Instructor: Anne Benoit

Visiting Associate Professor, CSE

Based on slides by Bistra Dilkina, Jennifer Welch, George Bebis, and Kevin Wayne

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

KT8.5 SEQUENCING PROBLEMS

Directed Hamiltonian Cycle

- DIR-HAM-CYCLE: given a digraph G = (V, E), does there exists a simple directed cycle Γ that contains every node in V?

- HAM-CYCLE: given an undirected graph G = (V, E), does there exists a simple cycle Γ that contains every node in V?

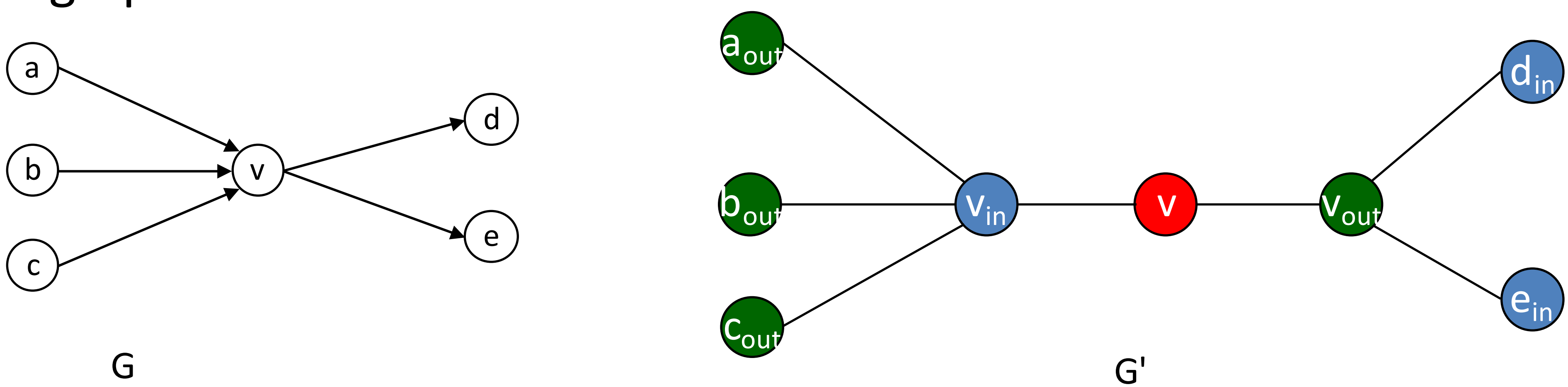
- DIR-HAM-CYCLE (HAM-CYCLE) is in NP
  - Certificate: Sequence of vertices
  - Certifier: Check if every vertex (except the first) appears exactly once, and that consecutive vertices are connected by a directed (undirected) edge

Directed Hamiltonian Cycle

- DIR-HAM-CYCLE: given a digraph G = (V, E), does there exists a simple directed cycle Γ that contains every node in V?

- Claim. DIR-HAM-CYCLE $\leq_P$ HAM-CYCLE.

- Pf. Given a directed graph G = (V, E), construct an undirected graph G' with 3n nodes.

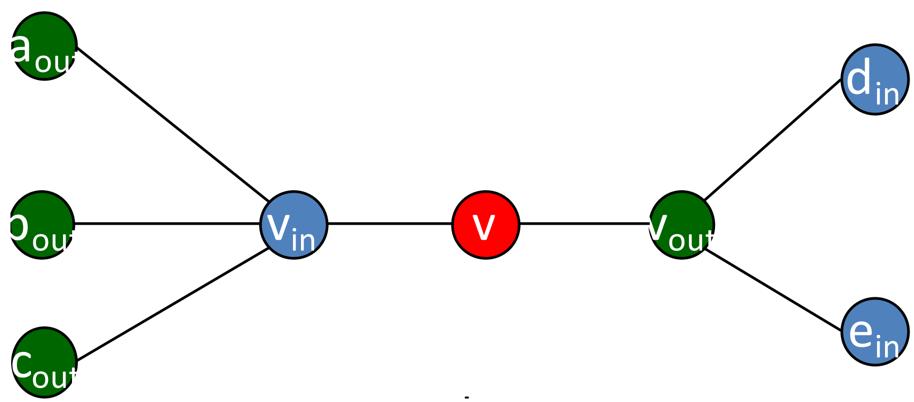


Directed Hamiltonian Cycle

- Claim. G has a Hamiltonian cycle iff G' does.
- Pf. $\Rightarrow$
  - Suppose G has a directed Hamiltonian cycle Γ.
  - Then G' has an undirected Hamiltonian cycle (same order).
- Pf. $\Leftarrow$
  - Suppose G' has an undirected Hamiltonian cycle Γ'.
  - Γ' must visit nodes in G' using one of following two orders:

    …, B, G, R, B, G, R, B, G, R, B, …

    …, B, R, G, B, R, G, B, R, G, B, …
  - Blue nodes in Γ' make up directed Hamiltonian cycle Γ in G, or reverse of one.



3-SAT Reduces to DIR-HAM-CYCLE

- Claim. 3-SAT $\leq_P$ DIR-HAM-CYCLE.

- Pf. Given an instance Φ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamiltonian cycle iff Φ is satisfiable.

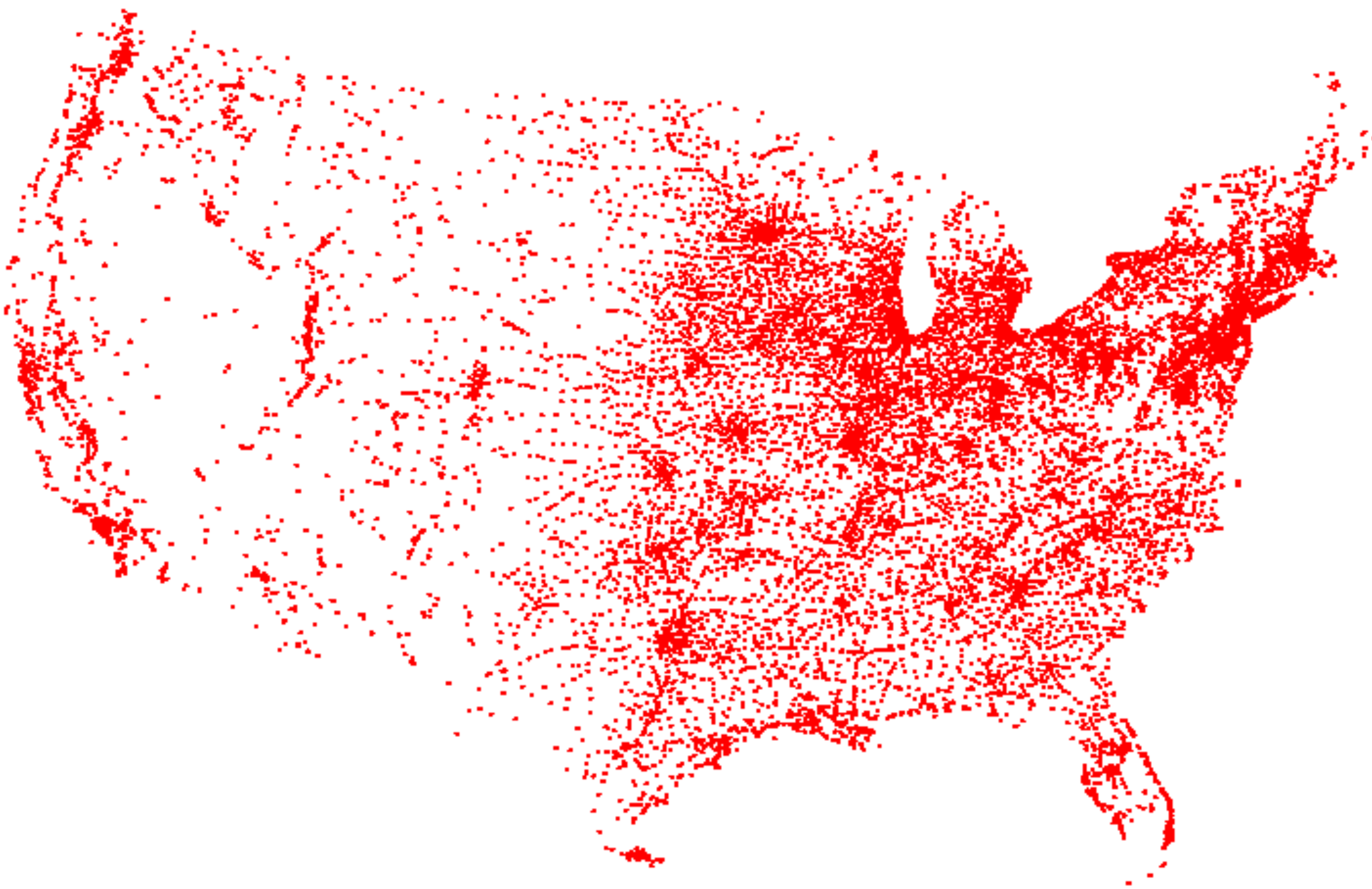
- Construction. First, create graph that has 2^n Hamiltonian cycles which correspond in a natural way to 2^n possible truth assignments. See KT8.5.

- See also CLRS34.5 for the reduction

  VERTEX-COVER $\leq_P$ HAM-CYCLE.

Traveling Salesperson Problem

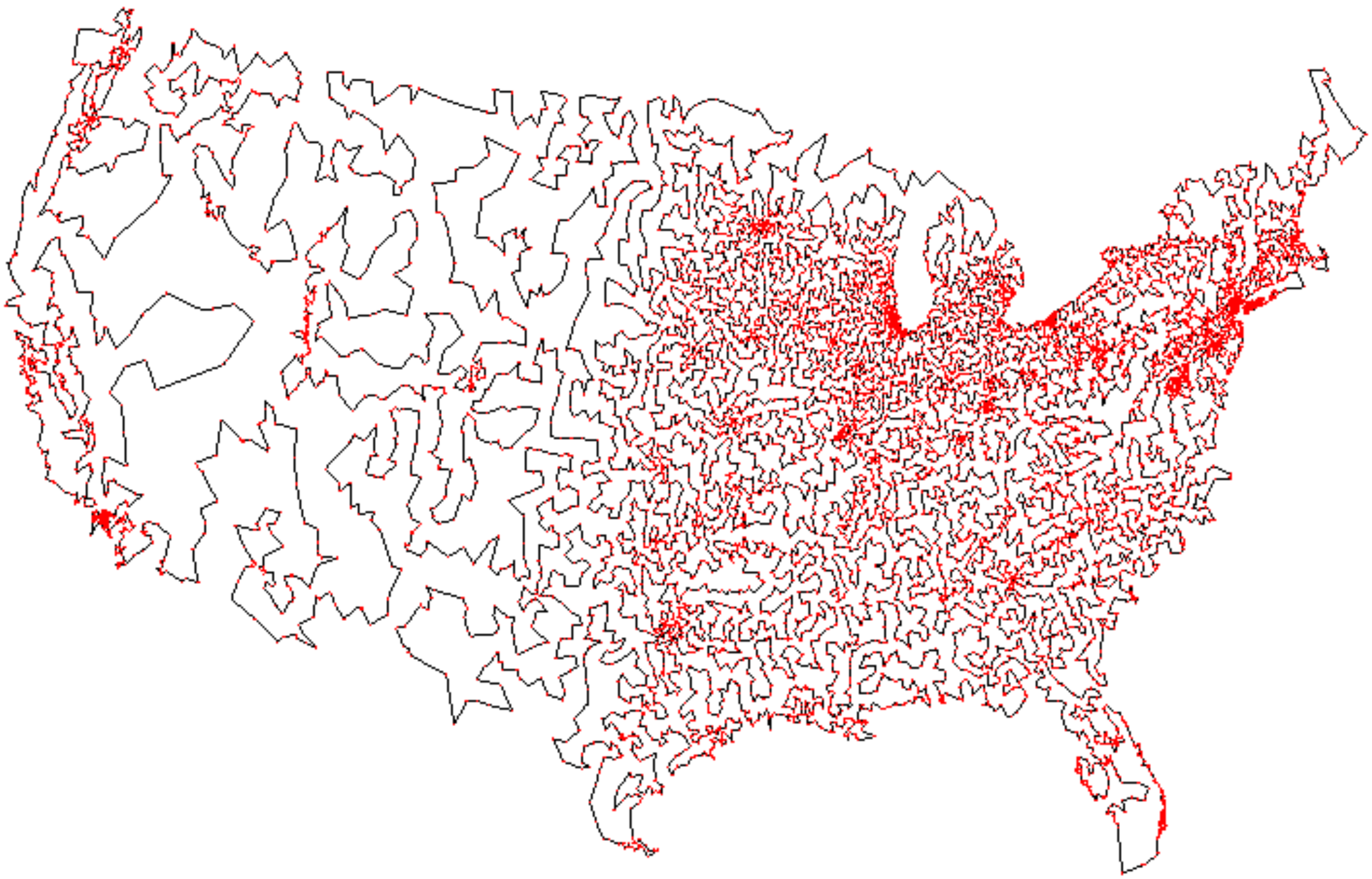
- TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length ≤ D?



All 13,509 cities in US with a population of at least 500
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

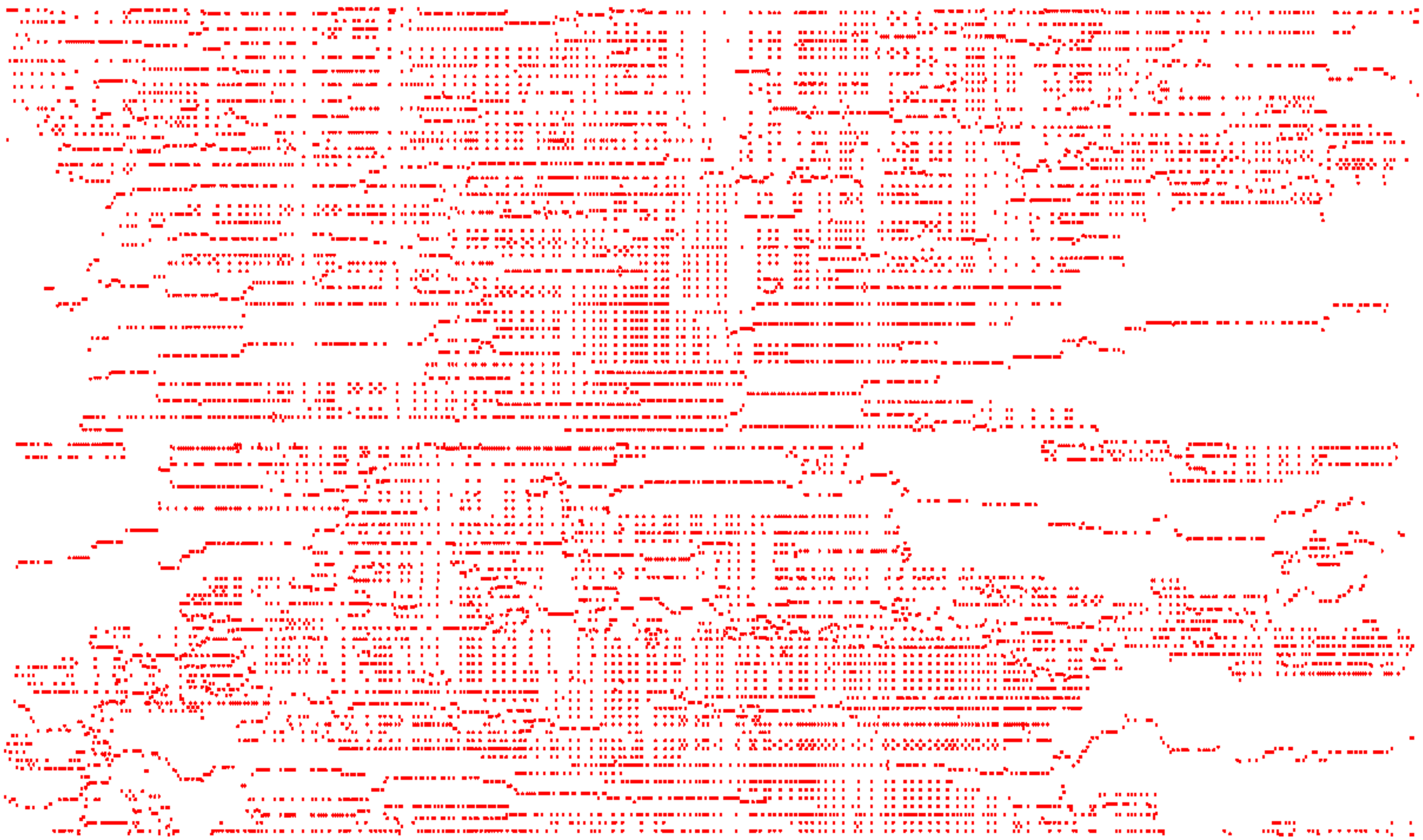
- TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length ≤ D?



Optimal TSP tour
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

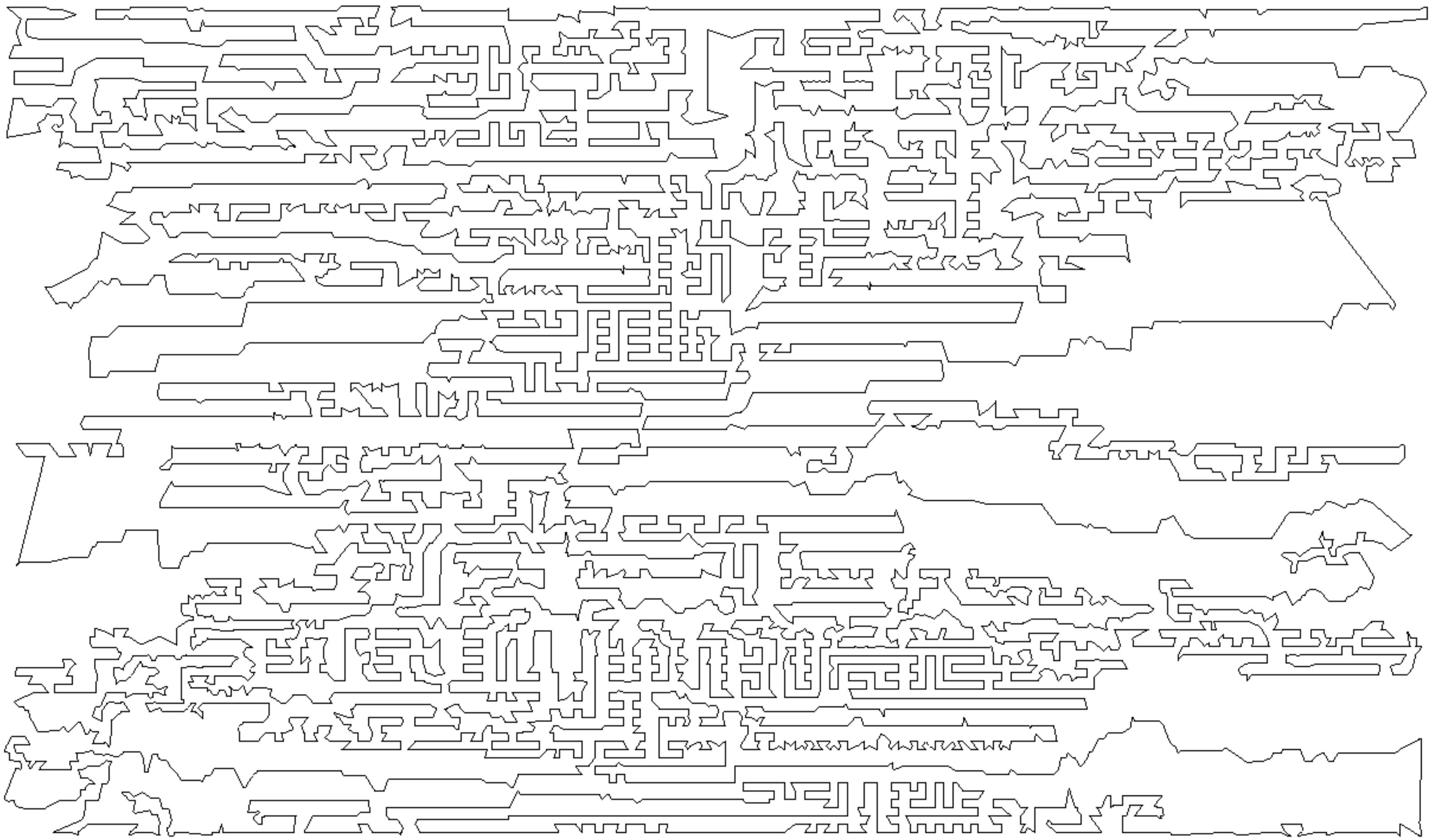
- TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length ≤ D?



11,849 holes to drill in a programmed logic array
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

- TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length ≤ D?



Optimal TSP tour
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

- TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length ≤ D?
- HAM-CYCLE: given a graph G = (V, E), does there exists a simple cycle that contains every node in V.

- Claim. HAM-CYCLE $\leq_P$ TSP.
- Pf.
  - Given instance I_1 of HAM-CYCLE: G = (V, E), create instance I_2 of TSP: n=|V| cities, distance function d, and D=n, with

$$d(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{if } (u, v) \notin E \end{cases}$$

  - G is Hamiltonian (I_1 has a solution) ⇔ TSP instance has a tour of length ≤ n (I_2 has a solution).

- Remark. TSP instance in reduction satisfies Δ-inequality.

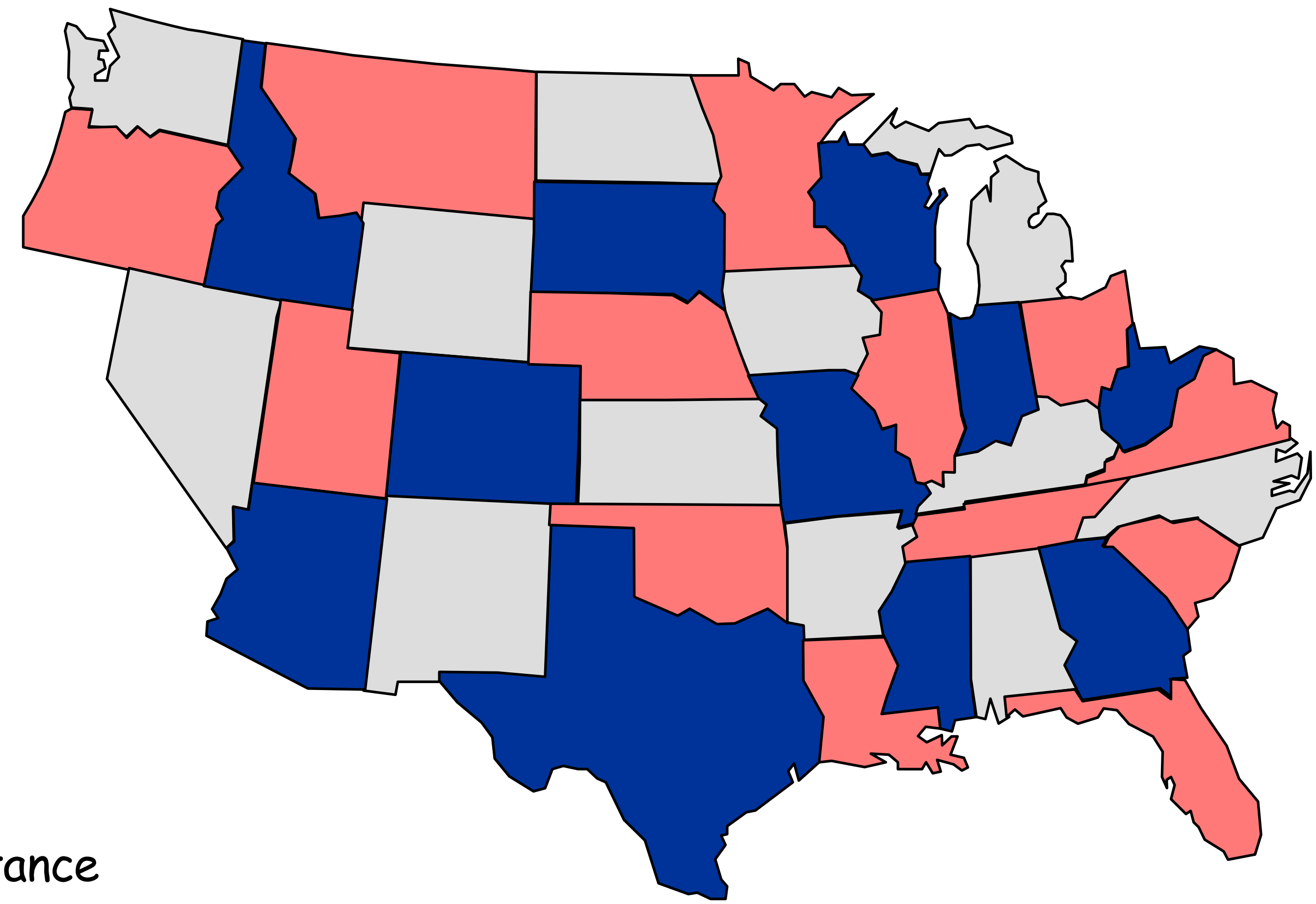
Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

KT8.6 PARTITIONING PROBLEMS
KT8.7 GRAPH COLORING

Map Coloring

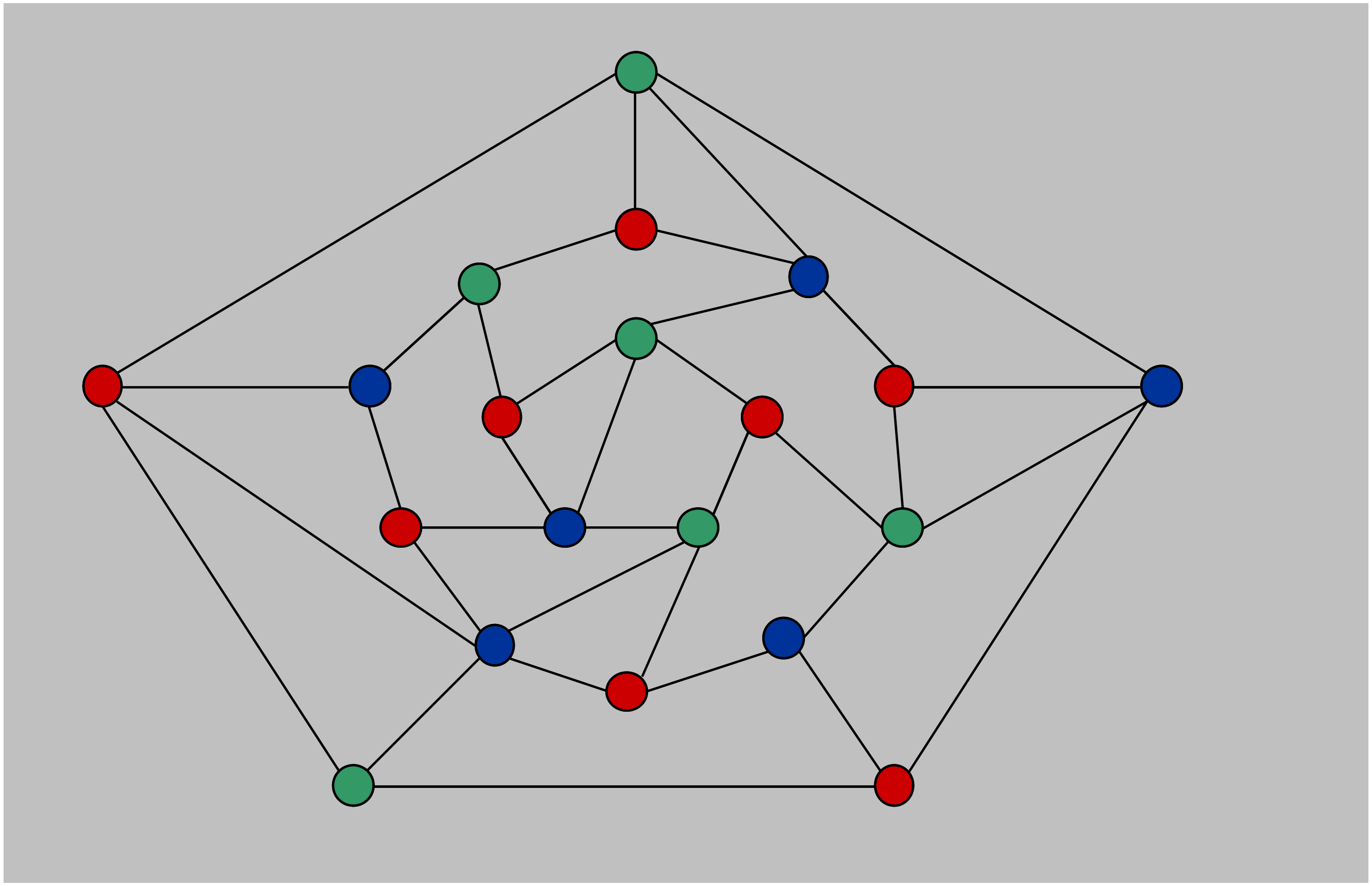
- Given a map, can it be colored using 3 colors so that no adjacent regions have the same color?



YES instance

Graph Coloring

- Each region is represented by a node in a graph; if 2 regions have a common boundary represent this by an edge between them. So we wish to assign colors to the nodes so that no two nodes have the same color if there is an edge between them



Graph Coloring

- We seek to assign a color to each node of a graph G so that if (u, v) is an edge, then u and v are assigned different colors; and the goal is to do this while using the smallest set of colors
- A k-coloring of G is a function f: V $\rightarrow$ {1, 2, …, k} so that for every edge (u, v), we have $f(u) \neq f(v)$.
- If G has a k-coloring, we say that it is a k-colorable graph
- Decision version: Given a graph G and a bound k, does G have a k-coloring?

Graph Coloring - Complexity

- Interesting point: the 4 color conjecture for maps in the plane
  - Planar graphs and k=4
  - Outstanding problem for over a century
  - Resolved in 1976 by Appel and Haken

  - Structure of the proof was induction on the number of regions but the induction step involved nearly 2000 complicated cases and had to be carried out by a computer

  - The problem of finding a reasonably short, human readable proof still remains open

Applications: Scheduling

- Example: schedule final exams and, being very considerate, avoid having a student do more than one exam a day
- What is the least number of days you need to schedule all the exams?
- Input: the pairs of exams that share students

	c1	c2	c3	c4	c5	c6	c7
c1		*	*	*		*	*
c2	*		*				*
c3	*	*		*			
c4	*		*		*	*	
c5				*		*	
c6	*			*	*		*
c7	*	*				*	

courses 7 and 2 have at least one student in common

Applications: Scheduling

- n processes or “jobs” (exams) on a system that can run multiple jobs concurrently but certain pairs of jobs cannot be scheduled at the same time because they both need a certain resource (student); over next k time steps (exam days) schedule each process to run in at least one time step;

- Graph G: node represents process, edge represents conflict, k-coloring represents a conflict free schedule - all nodes colored j can be scheduled in time j

Applications: Register Allocation

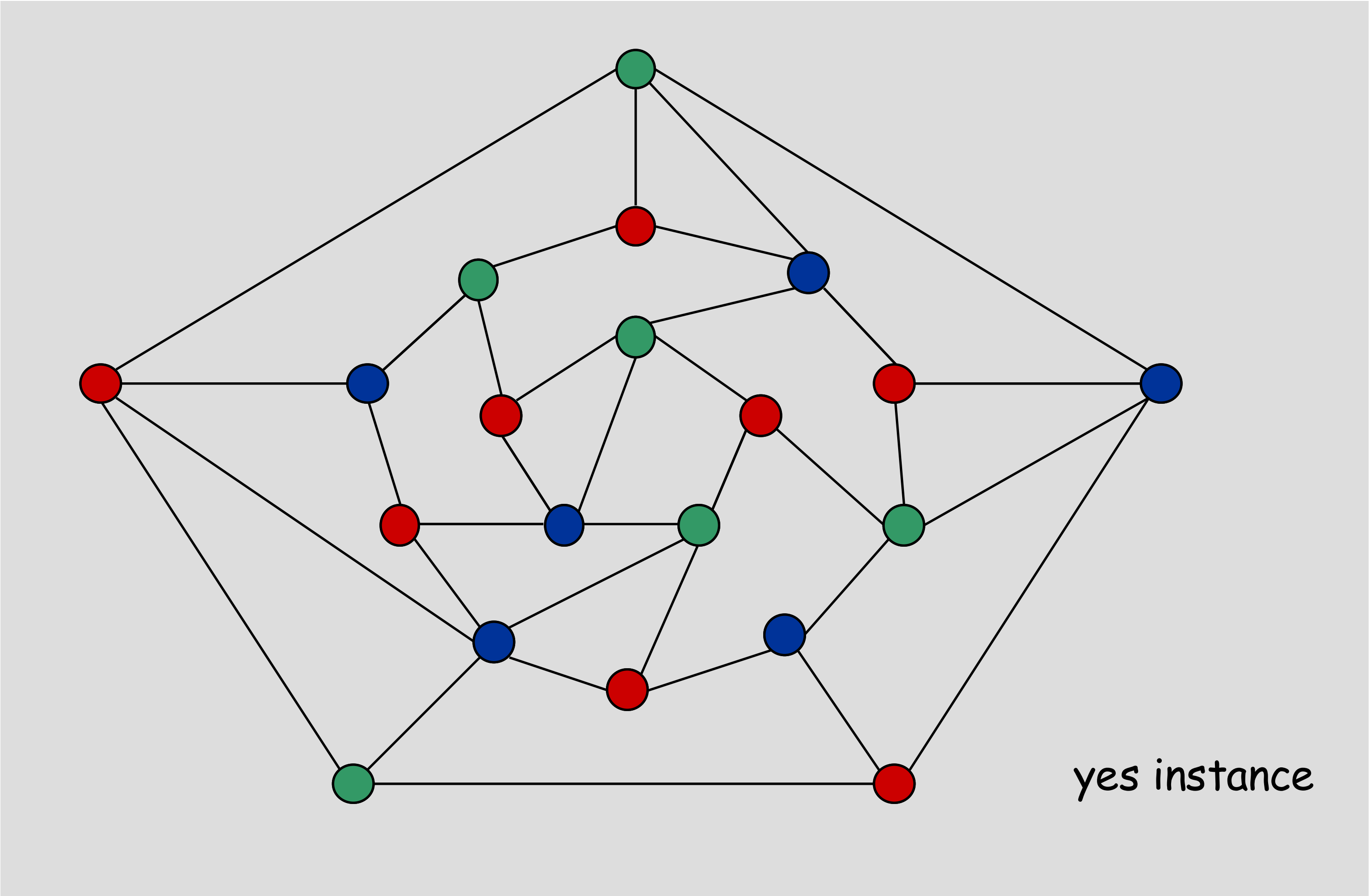
- Assign program variables to machine registers so that no more than k registers are used and no two program variables that are needed at the same time are assigned to the same register
  - Interference graph: Nodes are program variables names, edge between u and v if there exists an operation where both u and v are "live" at the same time
  - Observation: [Chaitin 1982] Can solve register allocation problem iff interference graph is k-colorable

Applications: Wavelength Assignment

- Wavelength assignment for wireless communication devices
- Assign one of k transmitting wavelengths to each of n devices
- 2 devices sufficiently close to each other have to be assigned different wavelengths to prevent interference
  - Interference graph: Nodes are devices, edge between u and v if devices are close enough to interfere with each other
  - Observation: Can solve wavelength assignment problem iff interference graph is k-colorable

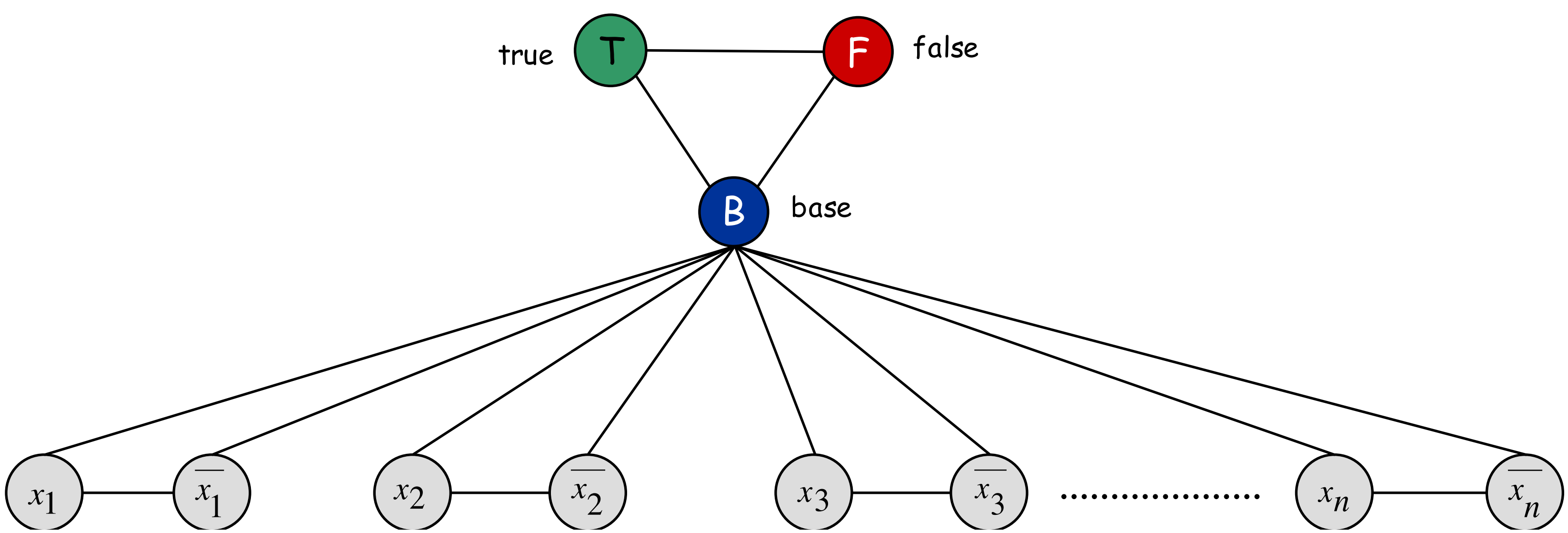
3-Colorability

- 3-COLOR: Given an undirected graph G does there exists a way to color the nodes using at most three colors(e.g. red, green, and blue) so that no adjacent nodes have the same color?

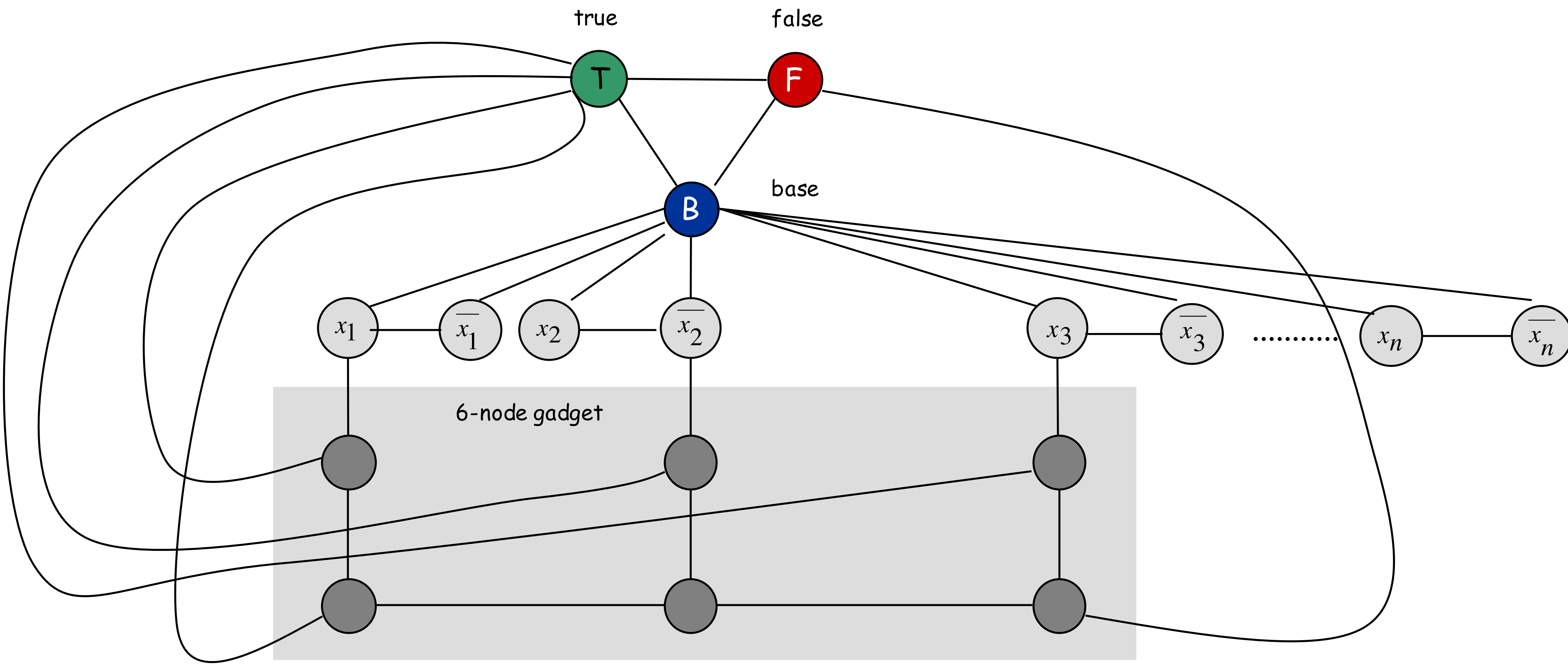


3-Colorability

- Claim: 3-SAT $\leq_P$ 3-COLOR
- Pf: Given 3-SAT instance Φ (I_1), we construct an instance I_2 of 3-COLOR that is 3-colorable iff Φ is satisfiable
- Construction
  - Create 3 new nodes T, F, B; connect them in a triangle (True, False, and Base)
  - For each literal, create a node
  - Connect each literal to B (i.e. every literal will have to be colored same as T or F)
  - Connect each literal to its negation (literals of same var cannot be same color)
  - For each clause, add gadget of 6 nodes and 13 edges as in next slide



3-Colorability



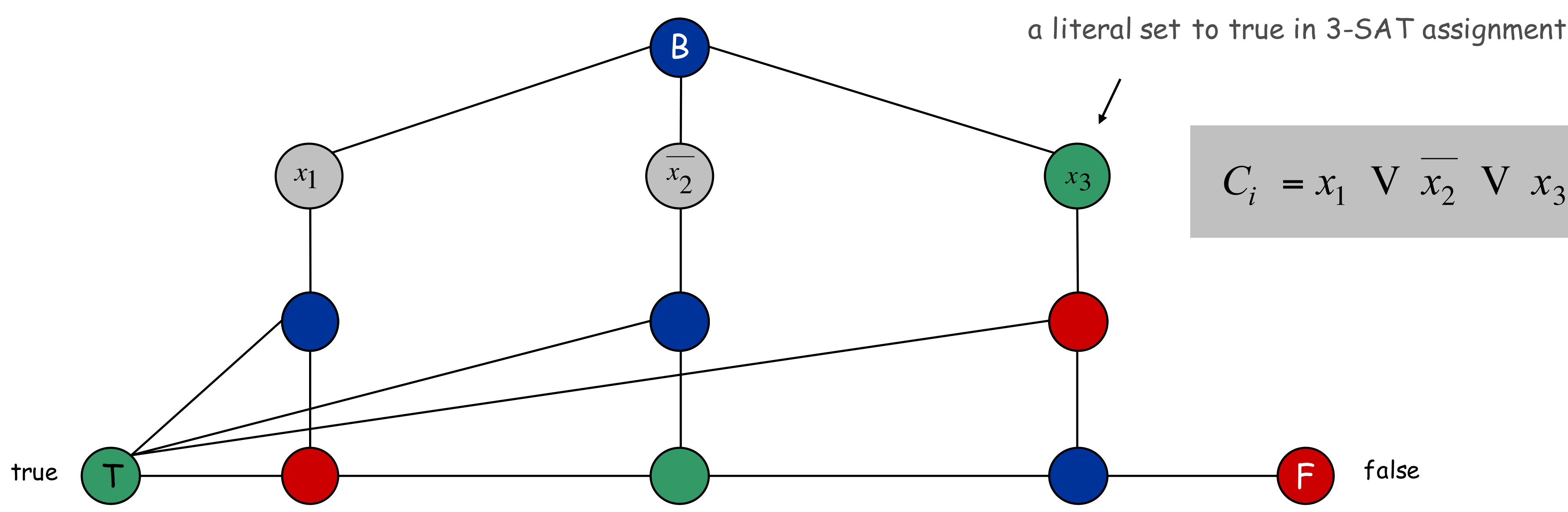
$$C_i = x_1 \vee \overline{x_2} \vee x_3$$

3-Colorability

Claim. Φ is satisfiable ($sol(I_1)$) iff graph is 3-colorable ($sol(I_2)$)

Pf. $\Rightarrow$ Suppose 3-SAT formula Φ is satisfiable.

- Color all nodes of true literals in the satisfying assignment with color T.
- Then at least one literal in each clause is true, hence colored T/green
- Color node below T/green node F/red, and node below that B/blue.
- Color remaining middle row nodes B/blue.
- Color remaining bottom nodes T/green or F/red as forced. ▪



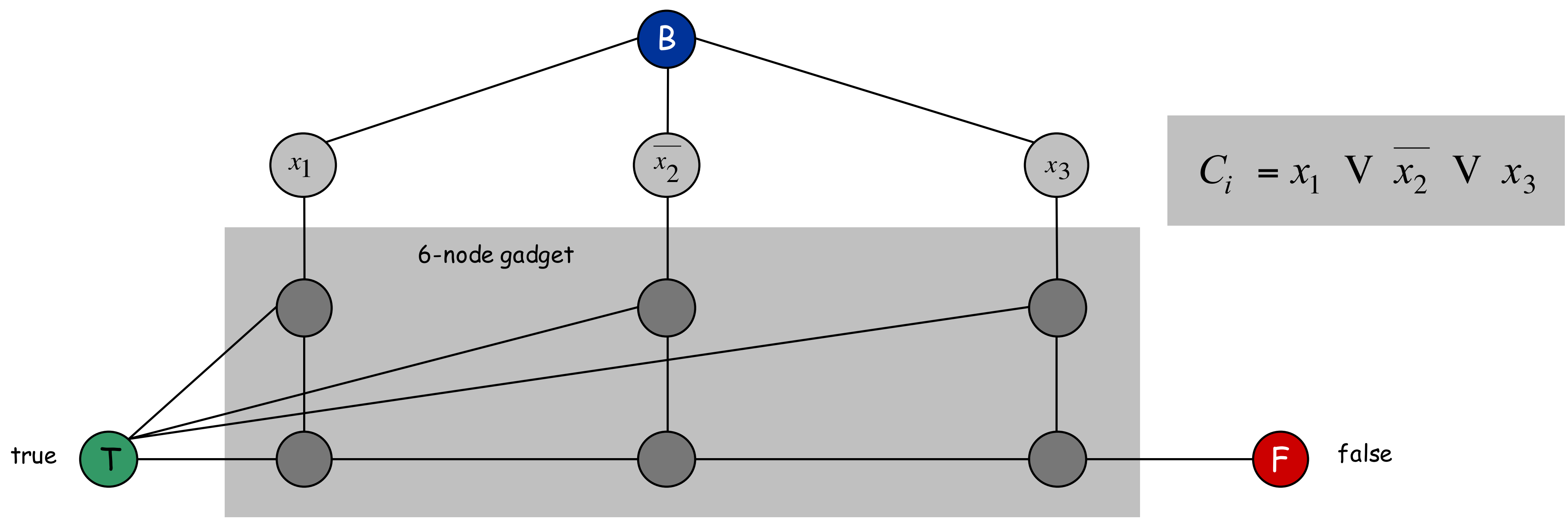
$$C_i = x_1 \vee \overline{x_2} \vee x_3$$

3-Colorability

Claim. Φ is satisfiable (sol(I_1)) iff graph is 3-colorable (sol(I_2))

Pf. $\Leftarrow$ Suppose graph is 3-colorable.

- Consider assignment that sets all T-colored literals to true.
- (i) ensures each literal is True or False.
- (ii) ensures a literal and its negation are opposites.
- (iii) ensures at least one literal in each clause is T (let's see why).



3-Colorability

Claim. Φ is satisfiable (sol(I_1)) iff graph is 3-colorable (sol(I_2))

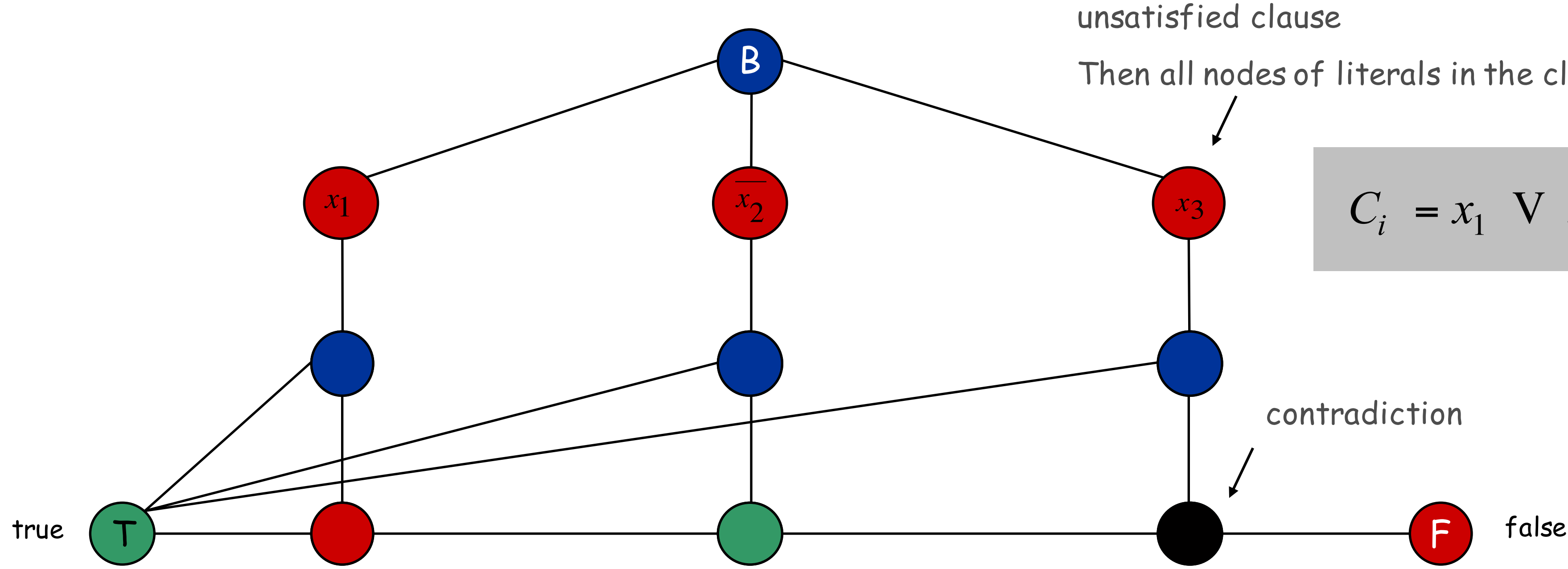
Pf. $\Leftarrow$ Suppose graph is 3-colorable.

- Consider assignment that sets all T-colored literals to true.
- (i) ensures each literal is True or False.
- (ii) ensures a literal and its negation are opposites.
- (iii) ensures at least one literal in each clause is T (let's see why).

For the sake of contradiction assume there is unsatisfied clause

Then all nodes of literals in the clause are red

$$C_i = x_1 \vee \overline{x_2} \vee x_3$$



Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

KT8.8 NUMERICAL PROBLEMS

Subset Sum

SUBSET-SUM. Given natural numbers $w_1, \ldots, w_n$ and an integer W, is there a subset that adds up to exactly W?

Ex: { 1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344 }, W = 3754.
Yes. 1 + 16 + 64 + 256 + 1040 + 1093 + 1284 = 3754.

Remark. With arithmetic problems, input integers are encoded in binary. Polynomial reduction must be polynomial in binary encoding.

Claim. 3-SAT $\leq_P$ SUBSET-SUM.

Subset Sum

Construction. Given 3-SAT instance Φ with n variables and k clauses, form 2n + 2k decimal integers, each of n+k digits, as illustrated below.

Claim. Φ is satisfiable iff there exists a subset that sums to W.
Pf. No carries possible.

$$C_1 = \bar{x} \vee y \vee z$$
$$C_2 = x \vee \bar{y} \vee z$$
$$C_3 = \bar{x} \vee \bar{y} \vee \bar{z}$$

	x	y	z	C_1	C_2	C_3	
x	1	0	0	0	1	0	100,010
¬ x	1	0	0	1	0	1	100,101
y	0	1	0	1	0	0	10,100
¬ y	0	1	0	0	1	1	10,011
z	0	0	1	1	1	0	1,110
¬ z	0	0	1	0	0	1	1,001
2k dummies to get clause columns to sum to 4	0	0	0	1	0	0	100
	0	0	0	2	0	0	200
	0	0	0	0	1	0	10
	0	0	0	0	2	0	20
	0	0	0	0	0	1	1
	0	0	0	0	0	2	2
W	1	1	1	4	4	4	111,444

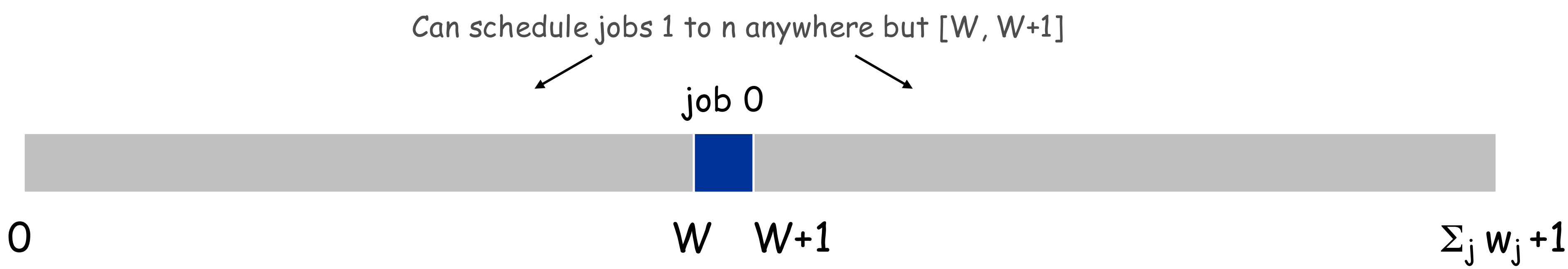
Scheduling With Release Times

SCHEDULE-RELEASE-TIMES. Given a set of n jobs with processing time t_i, release time r_i, and deadline d_i, is it possible to schedule all jobs on a single machine such that job i is processed with a contiguous slot of t_i time units in the interval $[r_i, d_i]$ between release time and deadline?

Claim. SUBSET-SUM $\leq_P$ SCHEDULE-RELEASE-TIMES.

Reduction. Given an instance of SUBSET-SUM $w_1, \ldots, w_n$, and target W,

- Create n jobs with processing time $t_i = w_i$, release time $r_i = 0$, and deadline $d_i = 1 + \Sigma_j w_j$.
- Create job 0 with $t_0 = 1$, release time $r_0 = W$, and deadline $d_0 = W+1$.



To get a feasible schedule, we have to be able partition the jobs exactly into 2 chunks, one of W and the other of $(\Sigma_j w_j - W)$ processing time.

Knapsack Problem

Knapsack problem.

- Given n objects and a "knapsack."
- Item i has value $v_i > 0$ and weighs $w_i > 0$. ← we'll assume $w_i \leq W$
- Knapsack can carry weight up to W.
- Goal: fill knapsack so as to maximize total value.

Ex: { 3, 4 } has value 40.

W = 11

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

Knapsack is NP-Complete

(Decision) KNAPSACK: Given a finite set X, nonnegative weights w_i, nonnegative values v_i, a weight limit W, and a target value V, is there a subset $S \subseteq X$ such that:

$$\begin{aligned} \sum_{i \in S} w_i &\leq W \\ \sum_{i \in S} v_i &\geq V \end{aligned}$$

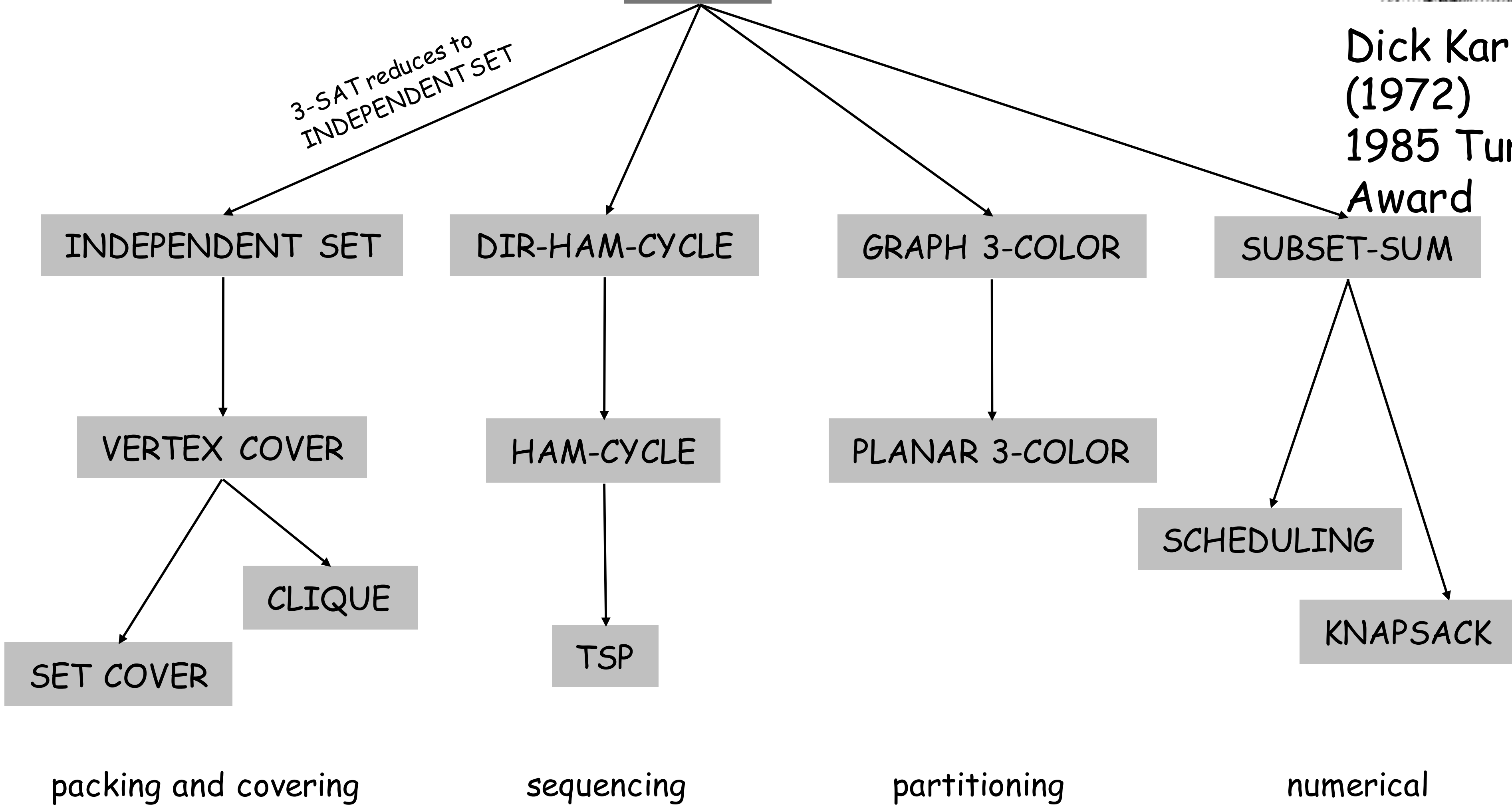
SUBSET-SUM: Given a finite set Y, nonnegative values u_i, and an integer U, is there a subset $S' \subseteq Y$ whose elements sum to exactly U?

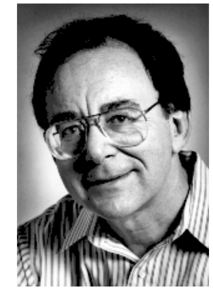
Claim. SUBSET-SUM $\leq_P$ KNAPSACK.

Reduction. Given instance $(u_1, \ldots, u_n, U)$ of SUBSET-SUM, create KNAPSACK instance:

$$\begin{aligned} v_i = w_i = u_i \qquad & \sum_{i \in S} u_i \leq U \\ V = W = U \qquad & \sum_{i \in S} u_i \geq U \end{aligned}$$

Polynomial-Time Reductions





Dick Karp
(1972)
1985 Turing
Award