Section 3

Efficiency and Hardness

Worst-case algorithmic complexity

- **Computational complexity theory:** *worst-case time/space taken by an algorithm to complete*
- **Algorithm $\mathcal{A}$**
  - e.g. to determine whether a graph $G = (V, E)$ is connected or not
  - input: G; size of input: $\nu = |V| + |E|$
- **How does the CPU time $\tau(\mathcal{A})$ used by $\mathcal{A}$ vary with ν?**
  - $\tau(\mathcal{A}) = O(\nu^k)$ for fixed k: polytime
  - $\tau(\mathcal{A}) = O(2^\nu)$: exponential
- **polytime $\leftrightarrow$ efficient**
- **exponential $\leftrightarrow$ inefficient**

Polytime algorithms are "efficient"

- Why are polynomials special?
- Many different variants of Turing Machines (TM)
- Polytime is *invariant* to all definitions of TM
- In practice, $O(\nu)$-$O(\nu^3)$ is an acceptable range covering most practically useful efficient algorithms
- Many exponential algorithms are also usable in practice for limited sizes

Instances and problems

- An input to an algorithm $\mathcal{A}$: *instance*
- Collection of all inputs for $\mathcal{A}$: *problem*
  consistent with "set of sentences" from decidability
- BUT:
  - A problem can be solved by different algorithms
  - There are problems which no algorithm can solve
- Given a problem P, what is the complexity of the *best algorithm* that solves P?

Complexity classes

- Focus on *decision problems*
- If $\exists$ polytime algorithm for P, then $P \in \mathbf{P}$
- If there is a polytime checkable *certificate* for all YES instances of P, then $P \in \mathbf{NP}$
- No-one knows whether $\mathbf{P} = \mathbf{NP}$ (we think not)
- NP includes problems for which we don't think a polytime algorithms exist

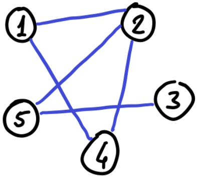
  e.g. k-CLIQUE, SUBSET-SUM, KNAPSACK, HAMILTONIAN CYCLE, SAT, ...

Subsection 1

Some combinatorial problems

k-CLIQUE

- <u>Instance:</u> $(G = (V, E), k)$
- <u>Problem:</u> determine whether G has a *clique* of size k



- 1-CLIQUE? YES (every graph is YES)
- 2-CLIQUE? YES (every non-empty graph is YES)
- 3-CLIQUE? YES (triangle $\{1, 2, 4\}$ is a certificate)
  certificate can be checked in $O(k) < O(n)$
- 4-CLIQUE? NO
  no polytime certificate unless $\mathrm{P} = \mathrm{NP}$

MP formulations for CLIQUE

Variables? Objective? Constraints?

MP formulations for CLIQUE

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rrcl} \forall \{i,j\} \not\in E & x_i + x_j & \le & 1 \\ & \sum\limits_{i \in V} x_i & = & k \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

MP formulations for CLIQUE

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rrcl} \forall \{i,j\} \notin E & x_i + x_j & \le & 1 \\ & \sum_{i \in V} x_i & = & k \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

- MAX CLIQUE:

$$\left.\begin{array}{rrcl} \max & \sum_{i \in V} x_i & & \\ \forall \{i,j\} \notin E & x_i + x_j & \le & 1 \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

SUBSET-SUM

- **Instance:** list $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$ **and** $b \in \mathbb{N}$
- **Problem: is there** $J \subseteq \{1, \ldots, n\}$ **such that** $\sum_{j \in J} a_j = b$**?**

- $a = (1, 1, 1, 4, 5)$, $b = 3$: **YES** $J = \{1, 2, 3\}$

  all $b \in \{0, \ldots, 12\}$ *yield YES instances*
- $a = (3, 6, 9, 12)$, $b = 20$: **NO**

MP formulations for SUBSET-SUM

Variables? Objective? Constraints?

MP formulations for SUBSET-SUM

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rcl}\sum\limits_{j\le n} a_j x_j & = & b \\ x & \in & \{0,1\}^n\end{array}\right\}$$

KNAPSACK

- **Instance:** $c, w \in \mathbb{N}^n$, $K \in \mathbb{N}$
- **Problem: find** $J \subseteq \{1, \ldots, n\}$ **s.t.** $c(J) \leq K$ **and** $w(J)$ **is maximum**

- $c = (1, 2, 3), w = (3, 4, 5), K = 3$
  - $c(J) \leq K$ **feasible for** J **in** $\varnothing, \{j\}, \{1, 2\}$
  - $w(\varnothing) = 0, w(\{1, 2\}) = 3 + 4 = 7, w(\{j\}) \leq 5$ **for** $j \leq n$

    $\Rightarrow J_{\mathsf{max}} = \{1, 2\}$
- $K = 0$**: infeasible**

- natively expressed as an optimization problem
- *notation:* $c(J) = \sum_{j \in J} c_j$ *(similarl for* $w(J)$*)*

MP formulation for KNAPSACK

Variables? Objective? Constraints?

MP formulation for KNAPSACK

Variables? Objective? Constraints?

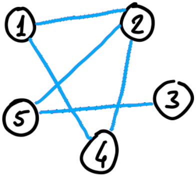
$$\left.\begin{array}{rcl} \max & \displaystyle\sum_{j\le n} w_j x_j & \\ & \displaystyle\sum_{j\le n} c_j x_j \le & K \\ & x \in & \{0,1\}^n \end{array}\right\}$$

Hamiltonian Cycle

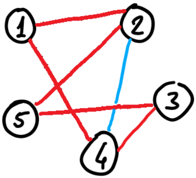
- **Instance:** $G = (V, E)$
- **Problem: does G have a *Hamiltonian cycle*?**

  cycle covering every $v \in V$ exactly once

NO



YES (cert. $1 \to 2 \to 5 \to 3 \to 4 \to 1$)



MP formulation for HAMILTONIAN CYCLE

Variables? Objective? Constraints?

MP formulation for HAMILTONIAN CYCLE

Variables? Objective? Constraints?

$$\forall i \in V \quad \sum_{\substack{j \in V \\ \{i,j\} \in E}} x_{ij} = 1 \tag{1}$$

$$\forall j \in V \quad \sum_{\substack{i \in V \\ \{i,j\} \in E}} x_{ij} = 1 \tag{2}$$

$$\forall \varnothing \subsetneq S \subsetneq V \quad \sum_{\substack{i \in S, j \notin S \\ \{i,j\} \in E}} x_{ij} \geq 1 \tag{3}$$

***WARNING*: second order statement!**

quantified over sets

other warning: need arcs not edges in (1)-(3)

Satisfiability (SAT)

- **Instance: open boolean logic sentence f in CNF**

$$\bigwedge_{i \le m} \bigvee_{j \in C_i} \ell_j$$

where $\ell_j \in \{x_j, \bar{x}_j\}$ for $j \le n$

- **Problem: is there $\phi : x \to \{0,1\}^n$ s.t. $\phi(f) = 1$?**

- $f \equiv (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2)$

  $x_1 = x_2 = 1, x_3 = 0$ **is a YES certificate**

- $f \equiv (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$

ϕ	$x = (1,1)$	$x = (0,0)$	$x = (1,0)$	$x = (0,1)$
false	C_2	C_1	C_3	C_4

MP formulation for SAT

Exercise

Subsection 2

NP-hardness

NP-Hardness

- Do *hard* problems exist? Depends on $\mathbf{P} \neq \mathbf{NP}$
- Next best thing: define *hardest problem in* NP
- A problem P is NP-*hard* if
  Every problem Q in NP can be solved in this way:
  1. given an instance q of Q transform it in polytime to an instance $\rho(q)$ of P s.t. q is YES iff $\rho(q)$ is YES
  2. run the best algorithm for P on $\rho(q)$, get answer $\alpha \in \{\text{YES}, \text{NO}\}$
  3. return α

  ρ is called a *polynomial reduction* from Q to P
- If P is in NP and is NP-hard, it is called NP-*complete*
- Every problem in NP reduces to SAT [Cook 1971]

Cook's theorem

Theorem 1: If a set S of strings is accepted by some nondeterministic Turing machine within polynomial time, then S is P-reducible to {DNF tautologies}.

Boolean decision variables store TM dynamics

Proposition symbols:

$P^i_{s,t}$ for $1 \le i \le \ell$, $1 \le s,t \le T$. $P^i_{s,t}$ is true iff tape square number s at step t contains the symbol σ_i.

Q^i_t for $1 \le i \le r$, $1 \le t \le T$. Q^i_t is true iff at step t the machine is in state q_i.

$S_{s,t}$ for $1 \le s,t \le T$ is true iff at time t square number s is scanned by the tape head.

Definition of TM dynamics in CNF

B_t asserts that at time t one and only one square is scanned:

$$B_t = (S_{1,t} \vee S_{2,t} \vee \ldots \vee S_{T,t}) \,\&\, [\mathop{\&}_{1 \le i < j \le T} (\neg S_{i,t} \vee \neg S_{j,t})]$$

$G^t_{i,j}$ asserts that if at time t the machine is in state q_i scanning symbol σ_j, then at time $t+1$ the machine is in state q_k, where q_k is the state given by the transition function for M.

$$G^t_{i,j} = \mathop{\&}_{s=1}^{T} (\neg Q^i_t \vee \neg S_{s,t} \vee \neg P^j_{s,t} \vee Q^k_{t+1})$$

Description of a dynamical system using a declarative programming language (SAT) — what MP is all about!