December 10

- How to prove $\mathcal{NP}$-completeness by reduction from known $\mathcal{NP}$-complete problem
- BIP-DECISION is $\mathcal{NP}$-complete
- SUBSET-SUM-DECISION is $\mathcal{NP}$-complete
- $\mathcal{NP}$-hard and weakly $\mathcal{NP}$-hard problems
- Consequences of $\mathcal{P} = \mathcal{NP}$ or $\mathcal{P} \neq \mathcal{NP}$

Presentation differs from Wolsey

- Wolsey only assumes SATISFIABILITY $\in \mathcal{NPC}$
- Wolsey proves BIP-DECISION $\in \mathcal{NPC}$

Proving $\mathcal{NP}$-completeness

A problem Q is $\mathcal{NP}$-complete if

(1) $Q \in \mathcal{NP}$

(2) $\forall R \in \mathcal{NP} : R \leq_p Q$

Proving (2) is difficult, but if we already know some $\mathcal{NP}$-complete problem it is easier

To prove Q is $\mathcal{NP}$-complete

(a) show $Q \in \mathcal{NP}$

(b) choose a problem $S \in \mathcal{NPC}$

(c) show that $S \leq_p Q$

Proof: We have shown (1) in (a), and (2) is valid since

$$\forall R \in \mathcal{NP} : R \leq_p S \leq_p Q$$

$\mathcal{NP}$-hard and weakly $\mathcal{NP}$-hard problems

A problem Q is $\mathcal{NP}$-complete if

(1) $Q \in \mathcal{NP}$

(2) $\forall R \in \mathcal{NP} : R \leq_p Q$

If only (2) holds we call problem $\mathcal{NP}$-hard.

An *optimization* problem is called $\mathcal{NP}$-hard if corresponding decision problem is $\mathcal{NP}$-complete.

A problem is *weakly* $\mathcal{NP}$-hard if it is $\mathcal{NP}$-hard but it can be solved in time polynomial in n and M, where n is length of input, and M is magnitude of coefficients.

Binary integer programming

Integer programming plays a central role in combinatorial optimization and operations research. Hence we will prove that

$$\text{BIP-DECISION} \in \mathcal{NPC}$$

Binary integer programming in decision form is defined as

$$\text{BIP-DECISION}(A, b, c, k) = \left\{ \begin{array}{lll} \max & \sum_{j=1}^{n} c_j x_j \geq k & \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, & i = 1, \ldots, m \\ & x_j \in \{0, 1\} & \end{array} \right\}$$

without loss of generality we assume A, b, c integers

Equivalent formulation

$$\text{BIP-DECISION}(A, b, c, k) = \text{feasible} \left\{ \begin{array}{ll} \sum_{j=1}^{n} -c_j x_j \leq -k & \\ \sum_{j=1}^{n} a_{ij} x_j \leq b_i, & i = 1, \ldots, m \\ x_j \in \{0, 1\}, & j = 1, \ldots, n \end{array} \right\}$$

or simply introducing new vaules of A, b, m

$$\text{BIP-DECISION}(A,b) = \\ \text{feasible}\left\{ \begin{array}{ll} \sum_{j=1}^{n} a_{ij}x_j \le b_i, & i = 1,\dots,m \\ x_j \in \{0,1\}, & j = 1,\dots,n \end{array} \right\}$$

Since $b - \Sigma_{j=1}^{n} a_{ij}x_j$ is bounded by some u_i we may add a sequence of slack variables $x'_1 + 2x'_2 + 4x'_3 \dots$ which spans $0 \dots, u_i$. Hence for new A, b, m, n, x

$$\text{BIP-DECISION}(A,b) = \\ \text{feasible}\left\{ \begin{array}{ll} \sum_{j=1}^{n} a_{ij}x_j = b_i, & i = 1,\dots,m \\ x_j \in \{0,1\}, & j = 1,\dots,n \end{array} \right\}$$

Step (a)

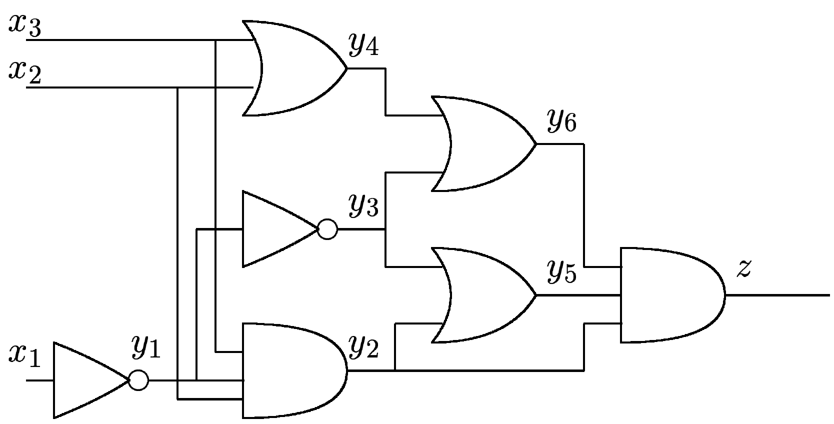
Certificate x. Check in polynomial time by checking if lefthand-side equals righthand-side.

Step (b)

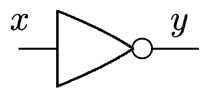
Reduce from CIRCUIT-SAT $\in \mathcal{NPC}$

step (c)

Given instance of CIRCUIT-SAT. Assign labels to output from each gate.

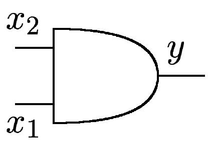


NOT-gate (x, y)



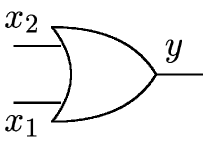
$$x + y \leq 1$$
$$x + y \geq 1$$

AND-gate (x_1, x_2, y)



$$x_1 + x_2 - 2y \geq 0$$
$$x_1 + x_2 - y \leq 1$$

OR-gate (x_1, x_2, y)



$$x_1 + x_2 - y \geq 0$$
$$x_1 + x_2 - 2y \leq 0$$

The final result should be true, add constraint

$$z = 1$$

All variables are boolean, add constraint

$$x_i, y_i, z \in \{0, 1\}$$

which completes the binary IP-model.

Check that "yes" instance is transformed to "yes" instance

$$\text{CIRCUIT-SAT}(X) = 1 \quad \Leftrightarrow \quad \text{BIP-DECISION}(f(X)) = 1$$

The resulting BIP model has $2n$ constraints, the same amount of variables. Reduction (conversion) can be done i polynomial time and space.

Subset-Sum is $\mathcal{NP}$-complete

Given n weights $w_1, \ldots, w_n$. Is it possible to choose a subset of the weights so that their weight sum equals c?

$$\text{SUBSET-SUM-DECISION}(w, c) = \text{feasible}\left\{ \begin{array}{l} \sum_{j=1}^{n} w_j x_j = c \\ x_j \in \{0, 1\} \end{array} \right\}$$

Outline of proof

(a) SUBSET-SUM-DECISION is in $\mathcal{NP}$.

(b) Choose BIP-DECISION.

(b) Prove BIP-DECISION $\leq_p$ SUBSET-SUM-DECISION

Reduction from BIP-DECISION

BIP-DECISION with two constraints

$$\text{BIP2-DECISION}(A, b) = \text{feasible}\left\{\begin{array}{l} \sum_{j=1}^{n} a_{1j} x_j = b_1 \\ \sum_{j=1}^{n} a_{2j} x_j = b_2 \\ x_j \in \{0,1\}, j = 1, \ldots, n \end{array}\right\}$$

Difference between the right and left side of the constraints

$$g(x) = b_1 - \sum_{j=1}^{n} a_{1j} x_j,$$

$$h(x) = b_2 - \sum_{j=1}^{n} a_{2j} x_j,$$

Bound on $g(x)$

$$g_{\min} = b_1 - \sum_{j=1}^{n} \overline{a}_j \leq g(x) \leq b_1 - \sum_{j=1}^{n} \underline{a}_j = g_{\max}$$

where $\overline{a}_j = \max\{a_{1j}, 0\}$ and $\underline{a}_j = \min\{a_{1j}, 0\}$.

$$\lambda > \max\left\{|g_{\max}|, |g_{\min}|\right\},$$

then

$$|g(x)| < \lambda$$

Multiply second constraint in BIP-DECISION with λ and add to first constraint

$$\text{BIP1-DECISION}(A, b) = \text{feasible} \left\{ \begin{array}{l} \sum_{j=1}^{n} (a_{1j} + \lambda a_{2j}) x_j = b_1 + \lambda b_2 = \tilde{b} \\ x_j \in \{0, 1\}, j = 1, \ldots, n \end{array} \right\}$$

which is a SUBSET-SUM-DECISION problem

Equivalence

Check that "yes" instance is transformed to "yes" instance

$$\text{BIP2-DECISION} = \text{"yes"} \quad \Leftrightarrow$$
$$\text{BIP1-DECISION} = \text{"yes"}$$

Proof

- Assume BIP2-DECISION(X)="yes" then obviously BIP1-DECISION(X)="yes".
- Assume BIP1-DECISION(X)="yes". Constraint in BIP1-DECISION may be written

  $$g(x) + \lambda h(x) = 0$$

  where

  $$h(x) = K$$

  must have K integer, as coefficients in $h(x)$ integers, hence

  $$g(x) + \lambda K = 0$$

  Since $|g(x)| < \lambda$, and K integer we must have $K = 0$

  $$h(x) = 0 \qquad g(x) = 0$$

  so both constraints are satisfied

□

Time consumption

The only computation is determination of λ (linear time). Output of BIP1-DECISION in linear time.

Reduction of BIP-DECISION to SUBSET-SUM-DECISION

Assume that BIP-DECISION has m constraints. Use above algorihtm m times. Resulting BIP1-DECISION will have the same set of solutions. Reduction in polynomial time.

Surprising result as SUBSET-SUM-DECISION contains all NP-problems

What happens with the coefficients (experts)

BIP2-DECISION to BIP1-DECISION

- all coefficients A, b in BIP2-DECISION are bounded by M
- input size
$$L(X) = \Theta(n \log M)$$
as $2n + 2$ coefficients.
- $\lambda \leq nM$
- output coefficients bounded by nM^2
- output size
$$O(X) = \Theta(n \log(nM))$$
since $n + 1$ coefficients of size $\log(nM^2)$.

BIP-DECISION to SUBSET-SUM-DECISION

- assume BIP-DECISION has m constraints
- output coefficients bounded by $n^{m-1}M^m$

SUBSET-SUM-DECISION (experts)

- A problem is weakly $\mathcal{NP}$-hard if it can be solved in time polynomial in n and M
- SUBSET-SUM is weakly $\mathcal{NP}$-hard
- We have shown BIP-DECISION $\leq_p$ SUBSET-SUM-DECISION
- Does it mean that BIP-DECISION is weakly $\mathcal{NP}$-hard

More $\mathcal{NP}$-complete problems

- Having three $\mathcal{NP}$-complete problems at hand it is easier to prove $\mathcal{NP}$-completeness of other problems
- Garey and Johnson (1979) first compendium of $\mathcal{NP}$-complete problems
- Crescenzi and Kann (2002) up-to-date database of $\mathcal{NP}$-complete problems

  `http://www.nada.kth.se/ viggo/problemlist/compendium.html`

  link from home page

Consequences of $\mathcal{P} = \mathcal{NP}$ or $\mathcal{P} \neq \mathcal{NP}$

- Mathematician (optimist): prove $\mathcal{P} = \mathcal{NP}$
- Mathematician (pessimist): prove $\mathcal{P} \neq \mathcal{NP}$
- Mathematician (thoughtful): find efficient algorithm close to optimium
- Probabilist (thoughtful): find efficient algorithm which returns optimal solution with high probability
- Engineer: heuristic algorithm
- Struggelig professor: study each problem