NP-Hardness of Integer Linear Programming

CSCI 4113 / 6101

Reminder: P & NP

Decision problem: yes/no answer

Reminder: P & NP

Decision problem: yes/no answer

Formal language: $L \subseteq \Sigma^*$

Reminder: P & NP

$$\left.\begin{array}{ll}\text{Decision problem:} & \text{yes/no answer} \\ \text{Formal language:} & L \subseteq \Sigma^* \end{array}\right\} \text{The same thing}$$

Reminder: P & NP

Decision problem: yes/no answer

Formal language: $L \subseteq \Sigma^*$

The same thing

P: All language $\mathcal{L}$ that can be decided in polynomial time.

Reminder: P & NP

Decision problem: yes/no answer

Formal language: $L \subseteq \Sigma^*$

The same thing

P: All language $\mathcal{L}$ that can be decided in polynomial time.

NP: All languages that can be verified in polynomial time.

$$L \in NP \iff \exists L' \in P : x \in L \iff \exists y \in \Sigma^* : |y| \le |x|^c,\ xy \in L'$$

Reminder: NP-Hardness

Observation: $P \subseteq NP$.

Reminder: NP-Hardness

Observation: $P \subseteq NP$.

Central question: $P = NP$?

Reminder: NP-Hardness

Observation: $P \subseteq NP$.

Central question: $P = NP$?

A language (decision problem) is NP-hard if $L \in P$ implies $P = NP$.

Reminder: NP-Hardness

Observation: $P \subseteq NP$.

Central question: $P = NP$?

A language (decision problem) is NP-hard if $L \in P$ implies $P = NP$.

NP-hardness is evidence that L cannot be decided in polynomial time.

$$P \neq NP \Rightarrow L \notin P$$

Decision vs Optimization

Every optimization problem has a corresponding decision problem:

- MST ⇒ Is there a spanning tree of weight $\leq w$?
- Shortest paths ⇒ Is there a path from s to t of length $\leq \ell$?

Decision vs Optimization

Every optimization problem has a corresponding decision problem:

- MST $\Rightarrow$ Is there a spanning tree of weight $\leq w$?
- Shortest paths $\Rightarrow$ Is there a path from s to t of length $\leq \ell$?

Observation: If the decision problem corresponding to Π is NP-hard, then so is Π.

Reminder: Polynomial-Time Reductions

Polynomial-time reduction from L to L'

$$x \in \Sigma^* \longrightarrow \boxed{\;R\;} \longrightarrow y \in \Sigma^*$$

- $x \in L \iff y \in L'$
- $T_R(|x|) \leq |x|^c \quad (c = \text{const})$

Reminder: Polynomial-Time Reductions

Polynomial-time reduction from L to L'

$$x \in \Sigma^* \longrightarrow \boxed{\quad R \quad} \longrightarrow y \in \Sigma^*$$

- $x \in L \iff y \in L'$
- $T_R(|x|) \leq |x|^c \quad (c = \text{const})$

Lemma: If R is a polynomial-time reduction from L to L' and L is NP-hard, then L' is NP-hard.

$$\begin{tikzcd} L' \in P \arrow[r] \arrow[rr, bend right] & L \in P \arrow[r] & P = NP \end{tikzcd}$$

Reminder: Polynomial-Time Reductions

Lemma: If R is a polynomial-time reduction from L to L' and $L' \in P$, then $L \in P$.

Reminder: Polynomial-Time Reductions

Lemma: If R is a polynomial-time reduction from L to L' and $L' \in P$, then $L \in P$.

Reminder: Polynomial-Time Reductions

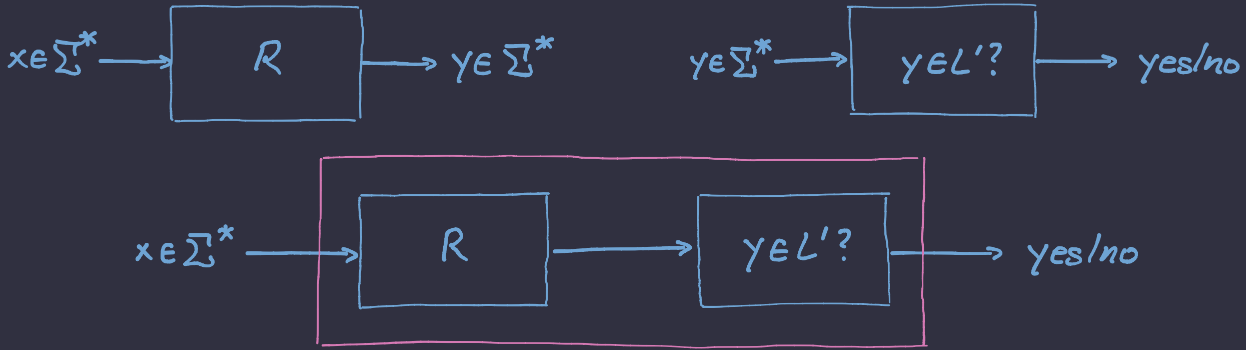
Lemma: If R is a polynomial-time reduction from L to L' and $L' \in P$, then $L \in P$.

$$x \in \Sigma^* \longrightarrow \boxed{R} \longrightarrow y \in \Sigma^* \qquad\qquad y \in \Sigma^* \longrightarrow \boxed{y \in L'?} \longrightarrow \text{yes/no}$$

$$x \in \Sigma^* \longrightarrow \boxed{R} \longrightarrow \boxed{y \in L'?} \longrightarrow \text{yes/no}$$

Reminder: Polynomial-Time Reductions

Lemma: If R is a polynomial-time reduction from L to L' and $L' \in P$, then $L \in P$.



Reminder: Polynomial-Time Reductions

Lemma: If R is a polynomial-time reduction from L to L' and $L' \in P$, then $L \in P$.

$$x \in \Sigma^* \longrightarrow \boxed{\quad R \quad} \longrightarrow y \in \Sigma^* \qquad\qquad y \in \Sigma^* \longrightarrow \boxed{\; y \in L'? \;} \longrightarrow \text{yes/no}$$

$$x \in \Sigma^* \longrightarrow \boxed{\boxed{\quad R \quad} \longrightarrow \boxed{\; y \in L'? \;}} \longrightarrow \text{yes/no}$$

$$T(|x|) \leq |x|^c + |y|^d \leq |x|^c + |x|^{cd} = O(|x|^{cd})$$

Satisfiability (SAT)

Conjunctive normal form (CNF)

$$F = \underbrace{(x_1 \vee x_2 \vee x_3)}_{} \wedge \underbrace{(x_1 \vee \overline{x_3} \vee x_4)}_{} \wedge \underbrace{(\overline{x_2} \vee \overline{x_4})}_{}$$

Clauses

- Variables: x_1, x_2, x_3, x_4
- Literals: $x_1, \overline{x_1}, \dots, x_4, \overline{x_4}$

Satisfiability (SAT)

Conjunctive normal form (CNF)

$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee \overline{x_4})$$

Satisfiability (SAT): Is there a truth assignment that makes F true?

$x_1 = \text{true}$ $\quad x_2 = \text{true}$ $\quad x_3 = \text{false}$ $\quad x_4 = \text{false}$

Satisfiability (SAT)

Conjunctive normal form (CNF)

$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee \overline{x_4})$$

Satisfiability (SAT): Is there a truth assignment that makes F true?

$x_1 = \text{true}$ $\quad x_2 = \text{true}$ $\quad x_3 = \text{false}$ $\quad x_4 = \text{false}$

Theorem: SAT is NP-hard.

A Polynomial-Time Reduction from SAT to ILP

$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee \overline{x_4})$$

$$\Downarrow$$

Π :

Lemma: F is satisfiable if and only if Π has a feasible solution.

A Polynomial-Time Reduction from SAT to ILP

$$F = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee \overline{x_4})$$

$$\Downarrow$$

Π :

Lemma: F is satisfiable if and only if Π has a feasible solution.

Corollary: ILP is NP-hard.

A Polynomial-Time Reduction from SAT to ILP

$$F = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_3} \lor x_4) \land (\overline{x_2} \lor \overline{x_4})$$

$$\Downarrow$$

Π : Maximize 0

s.t. $x_1 + x_2 + x_3 \geq 1$

$x_1 + (1 - x_3) + x_4 \geq 1$

$(1 - x_2) + (1 - x_4) \geq 1$

$x_1, x_2, x_3, x_4 \in \{0, 1\}$

Lemma: F is satisfiable if and only if Π has a feasible solution.

Corollary: ILP is NP-hard.