NP Complete Problems

Decision vs. Optimization problems

Algorithm \((I)\) \(\rightarrow\) actual solution, with the optimal cost

\[
\begin{align*}
\text{instance} & \quad \text{Input} \\
\text{Optimization} & \\
\end{align*}
\]

\[
\begin{align*}
\text{Alg}(I, b) & \quad \leftrightarrow \quad \text{Yes/True/1} \\
\text{Yes/True/1} & \quad \leftrightarrow \quad \text{No/False/0} \\
\end{align*}
\]

\[
\begin{align*}
\underbrace{\text{instance/}} & \quad \text{bound on the cost/value} \\
9+\text{poly} & \\
\end{align*}
\]

Decision problem

polynomial time REDUCTIONS

\[
\begin{align*}
P : \text{ poly time solvable} & \\
\end{align*}
\]

\[
\begin{align*}
\text{NP : (nondeterministic poly time)} & \\
\text{verified in poly time} & \\
\text{given a solution, we can check if it is correct in polynomial time} & \\
\end{align*}
\]
In polynomial time

- Easy for
decision
- Hard for
optimization

\[ d_{ij} = \text{dist}(i, j) \]

**Opt-TSP**: Given all the pair-wise distance between vertices,

return the optimal path that visits each vertex

**Do** \( \text{opt-TSP} \in \text{NP?} \)

\( \rightarrow \) Can we verify a solution in \( P \) time

given a solution \( S = (v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5) \)

\( \sum_{i \in S} c_{ij} \)

\( \text{Cost of TSP} \)

Is \( v \) optimal?

Looks like no efficient way of answering this in \( P \) time.

**c-TSP**: \( (T, S, b) \)

domain

Given \( S \), verification means: Is \( w(S) \leq b \)?

1) Compute \( w = \sum_{ij \in S} c_{ij} \)

\( O(|V| + |E|) \)

from \( S \)

2) Is \( w \leq b \)

\( \begin{cases} \text{YES} & \text{if} \quad w \leq b \\ \text{NO} & \text{otherwise} \end{cases} \)

\( O(1) \)

we can verify in \( P \) time.
Q1 \[ \text{d-Tsp:} \quad \frac{G}{\text{dij} \geq 0} \]

Is there a TSP with value \( v \leq b \)?

NP-Complete

Q2 \[ \text{opt-Tsp:} \quad \frac{G}{\text{dij}} \]

give actual TSP solution with optimal \( v \)

NP-hard

\[ \text{d-Tsp} \equiv \text{opt-Tsp} \]

poly-time reduction/ transformation

bit complexity of \( b \) \(-\) Important

transformation better be \( \log b \)

Reduce \( \text{d-Tsp} \) to \( \text{opt-Tsp} \)

\[ T = \{ \frac{G}{\text{dij}}, b \} \]

\[ \text{opt-Tsp} \]

\[ b \]

\[ \text{yes/no} \]

\( O(1) \) check
Using the opt-TSP to solve d-TSP is trivial O(1) time

How to use d-TSP to answer opt-TSP?
how can we find the optimal TSP value v, using only d-TSP:

\[ I = \{ s \mid d_{i,j} \} \]

\[ \text{opt-TSP} \]

\[ d = \text{TSP} \]

\[ \text{Optimal Value} \leq L \]

\[ \text{Yes} \]

\[ \text{No} \]

\[ \text{Use double binary search} \]

\[ O(\log L) \]

Actual Optimal Value for TSP

Naive:
\[ b = 1, b = 2, b = 3, \ldots, b = 100, b = 101 \]

No
No
No
No
No

\[ b \in \{1, 2, \ldots, 102\} \]

When we get the first Yes, that is the optimal value.

\[ \text{Check the value} \]

\[ b = 102 \]

\[ b = 102 = v \]

binary-search (double binary search)

\[ b = 2^0 = 1 \]
\[ b = 2^1 = 2 \]
\[ b = 2^2 = 4 \]

Unknown value v -

lecture23 Page 4
\[ b = 2^k \]

At most \( \lceil \log v \rceil \) steps to find the first interval.

b) \( v \in [2^{k-1}, 2^k] \)

Do a regular binary search on this interval to find the actual \( v \).

\[ 2^{k-1} \quad 2^k \]

Q: \( v \leq \text{mid} \)

No -

Only \( 2^{k-1} \) numbers

\( \log (2^{k-1}) = k-1 \) steps
\[ k = \log_2(n) \]

- **P**: poly-time solvable
- **NP**: poly-time verifiable
  \[ d = T^p \in NP \]
  \[ \text{opt} = T^{sp} \notin NP \]

**NP-Complete**: \( NP_c \)

- Harder problems in \( NP \)
- All these problems are "the same"
- Poly-time transformation from one to the other

**P** (or not)

\[ \Omega(?) = T^{sp} = O(2^n \cdot n^2) \text{ time} \]

**P**? or nor

**NP Complete Problems**:

A problem \( L \) belongs to \( NP \)-complete if

lecture23 Page 6
1) \( L \in \text{NP} \iff \text{verify in poly-time} \)

\[
\begin{align*}
\text{Equivalence of all problems in NP} &
\end{align*}
\]

2) for all problem \( L' \) in NP, \( L' \) can be \( \text{reduced} \) to \( L \), denoted as \( L' \rightarrow L \).

\( L' \rightarrow L \): \( L' \) can be poly-time reduced to \( L \).

- \( x \) is a solution to \( L' \) if and only if \( f(x) \) is a solution to \( L \).
- And \( f \) is computable in poly-time.
- \( f \): transformation function.

**SAT is NP Complete**

---

\( X = 0 \)

\( y = 1 \)

\( z = 1 \)

\[
\begin{align*}
\text{boolean satisfiability problem} &
\end{align*}
\]

\[
\begin{align*}
\left( x \lor y \lor \overline{z} \right) \land \left( \overline{x} \lor \overline{y} \right) \land \left( y \lor z \right)
\end{align*}
\]

- Clause
- Conjunction: AND

**SAT**: is the expression satisfiable?

Is there an assignment of 0/1 to the variables, such that expression is true?
1) \text{SAT} \in \text{NP} \\
In \text{O}(N) \text{ time we can verify a solution} \\
\[ N = \text{Sum of all clause lengths} \]
\[ N = \text{Poly in input size} \]

2) All problems \( \mathcal{L}' \) in \text{NP} reduce to \text{SAT} \\
\[ \xrightarrow{\text{poly-time transformation}} \]

\[ \mathcal{L}' \text{ can be coded as } \]
\[ \text{digital circuit (boolean expression)} \]
\[ \text{(hard coded \textless hard ware solution)} \]

\[ \text{d-TSP} \in \text{NP} \]
\[ \text{digital circuit} \rightarrow \text{boolean expression} \]

3) We have to show that the digital circuit is 
\[ \text{poly in size of } \mathcal{L}' \text{'s encoding} \]

\text{Proof exists!}

Instead of using \text{poly-size} digital circuits for all \text{NPC} problems.
Use reduction trick!

SAT \rightarrow d-TSP

How to show that a problem \( L' \) is \( NP \) complete?

1) \( L' \in NP \)

2) Find some other \( NPC \) problem \( L \), then show that \( L \rightarrow L' \)

\exists \text{ a poly-time transformation from } L \to L'

Original: \( \forall L \in NP \, L \rightarrow L' \)

Very difficult

\( SAT \rightarrow 3SAT \)

Known problem

\( NPC \leftarrow SAP \)
Show \( \exists \text{ SAT } \in \text{ NPC} \)

\[ \downarrow \]

Specialization of general \( \text{ SAT} \)

\[ \exists \text{ SAT } \leadsto \text{ no clause can have more than 3 literals} \]

\( \exists \text{ SAT } : \) \((x \lor y \lor z \lor \neg \bar{w} \lor \neg \bar{a}) \land (c) \land (c) \)

5 literals,

\[ \exists \text{ SAT } : \) \((x \lor y \lor z \lor \neg \bar{w} \lor \neg \bar{a}) \land (y \lor \bar{w}) \land (x \lor \bar{a}) \]

\[ |G| \leq 5 \]

Show \( \exists \text{ SAT } \in \text{ NPC} ? \)

1) \( \exists \text{ SAT } \in \text{ NP } \) \( \leadsto \) obviously true

Verify in \( \mathcal{O} \) time

2) \( \text{ SAT } \leadsto \exists \text{ SAT } \)

Reduction

\( \text{ SAT } \equiv \exists \text{ SAT } \)

\[ \Rightarrow \text{ poly time} \]

\((c \lor y \lor z \lor \neg w \lor \neg a) \land (c) \land (c) \)
\[ (c_1 \lor c_2) \lor (v_3 \lor v_4) \lor (v_5 \lor v_6) \land (\overline{y_1} \lor v_2 \lor y_2) \land (\overline{y_2} \lor v_4 \lor y_4) \land (\overline{y_3} \lor v_5 \lor v_6) \]

\[ y_1, y_2, y_3 : \text{dummy variables} \]