Strong NP-Completeness

Computability and Complexity
Andrei Bulatov

Knapsack

Knapsack

Instance: A sequence of positive integer values $V = \{v_1, v_2 \dots, v_n\}$
a set of positive integer weights $W = \{w_1, w_2, \dots, w_n\}$
a target value t, and a weight limit l.

Question: Is there a subset $T \subseteq S$ such that $\sum_{i\in T} v_i \ge t$ and $\sum_{i\in T} w_i \le t$?

Step 1: The problem Knapsack is in **NP**: the set T is the certificate

Step 2: To show that Knapsack is NP-complete we shall reduce SubsetSum to Knapsack

SubsetSum

Instance: A sequence of positive integers $S = \{a_1, \dots, a_n\}$ and a target integer t.

Question: Is there a subset $T \subseteq S$ such that $\sum_{i\in T} a_i = t$?

Given a SubsetSum instance, that is a set $S = \{a_1, a_2, \dots, a_n\}$ and a target number t

- Set $v_i = w_i = a_i$
- Set $t = l = t$

Then for any subset $T \subseteq S$

$\sum_{i\in T} a_i = t$ if and only if $\sum_{i\in T} v_i = \sum_{i\in T} a_i \ge t$ and $\sum_{i\in T} w_i = \sum_{i\in T} a_i \le t$

Polynomial Algorithm

Knapsack can be solved in $O(nl)$ time using dynamic programming

- Create a $(l+1) \times (n+1)$ matrix M
- Set $M(w,0)$ and $M(0,i)$ to 0 for all $w \in \{1,2,\dots,l\}$ and $i \in \{1,2,\dots,n\}$
- For $i = 1,2,\dots,n,$ set entry $M(w,i)$ as follows

  $$M(w, i+1) = \max\{M(w,i), M(w - w_{i+1}, i) + v_{i+1}\}$$

  ($M(w,i)$ is the largest total value obtainable by selecting from the first i items with weight limit w
- Answer yes if $M(l,n+1) \ge t,$ otherwise no

Example Construction

Let $V = \{1,3,4,2\}$, $W = \{1,1,3,2\}$, $t = 7$ and $l = 5$

w \ i	0	1	2	3	4
0	0	0	0	0	0
1	0	1	3	3	3
2	0	1	4	4	4
3	0	1	4	4	5
4	0	1	4	7	7
5	0	1	4	8	8

Pseudo-polynomial Time

This algorithm is not sufficient to show that Knapsack is in **P**!

The length of the input to Knapsack is in $O(n \log l)$ — so nl is not bounded by a polynomial function of the input length

- An algorithm which is polynomial in the size of the numbers in the input is called pseudo-polynomial
- NP-complete problem which remains NP-complete when all numbers in the input are bounded by some polynomial in the length of the input is called strongly NP-complete[1]

[1]See Garey and Johnson for more discussion of strong NP-completeness

Strongly NP-complete Problems

- Any NP-complete problem without numerical data is strongly NP-complete: SAT, Hamiltonian Circuit, other graph problems
- TSP(D) is strongly NP-complete

  (From the reduction of HamCircuit we know that even if the distances between cities are bounded by a linear polynomial, the problem remains equivalent to HamCircuit)
- SubsetSum?

  NO! It is a subproblem of Knapsack

NP and coNP

For a language L over an alphabet Σ, we denote $\overline{L}$ the complement of L, the language $\Sigma^* - L$

Definition
The class of languages L such that $\overline{L}$ has a polynomial time verifiers is called **coNP**

In other words, a problem belongs to coNP if the no-instances have succinct certificates

Examples

No Hamilton Circuit

Instance: A graph G.

Question: Is it true that G has no Hamiltonian circuit?

Prime (= Not Composite)

Instance: A positive integer k.

Question: Is k prime?

More Examples

Validity

Instance: A conjunctive normal form Φ.

Question: Is Φ valid?

NP-complete vs. coNP-complete

Definition
A language L is said to be **coNP**-complete if $L \in$ **coNP**, for any $A \in$ **coNP**, $A \le L$

Theorem
If L is NP-complete, then $\overline{L}$ is coNP-complete

Proof

Let L be NP-complete and $A \in$ **coNP**.

Then $\overline{A} \in$ **NP**, and therefore is poly-time reducible to L with a reduction R
To show that R is a reduction from A to $\overline{L}$, we just note that

$$x \in A \Leftrightarrow x \notin \overline{A} \Leftrightarrow R(x) \notin L \Leftrightarrow R(x) \in \overline{L}$$

Theorem

If a coNP-complete problem is in **NP**, then **NP** = **coNP**

Proof

If $L \in$ **NP** is coNP-complete, then every $A \in$ **coNP** is poly-time reducible to L. Since L belongs to **NP**, this means $A \in$ **NP**.

Conversely, if $A \in$ **NP**, then A is poly-time reducible to L. Since $L \in$ **coNP**, this means that $A \in$ **coNP**

Corollary

If **NP** $\neq$ **coNP,** then Validity, NoHamCircuit, etc. do not belong to **NP**

Around NP