NP-hardness (again, for the last time)

NP-hard = no polynomial-time algorithm (morally)

To prove $X$ is NP-hard:
1. Reduce known NP-hard problem $Y$ to $X$:
   - Describe an algorithm to transform arbitrary
     instance $y$ of $Y$ to special instance
     $x$ of $X$ so that

   - If $y$ is a "good" instance of $Y$,
     then $x$ is a "good" instance of $X$
   - If $y$ is a "bad" instance of $Y$
     then $x$ is a "bad" instance of $X$

![Diagram]

Typical proof:
- Show that any certificate that $y$ is good for $Y$
  becomes a certificate that $x$ is good for $X$
- Show that any certificate that $x$ is good for $X$
  must come from a certificate that $y$ is good for $Y$

This is the hard part!

Reduction must force structure on certificates for $X$
Pebbling Problem

Pebbling is a solitaire game played on an undirected graph $G$, where each vertex has zero or more pebbles. A single pebbling move consists of removing two pebbles from a vertex $v$ and adding one pebble to an arbitrary neighbor of $v$. (Obviously, the vertex $v$ must have at least two pebbles before the move.) The PebbleDestruction problem asks, given a graph $G = (V; E)$ and a pebble count $p(v)$ for each vertex $v$, whether there is a sequence of pebbling moves that removes all but one pebble. Prove that PebbleDestruction is NP-complete.

First, I show that it is in NP since I can verify the solution in polynomial time, tracing back the pebble count from just one pebble.

Next, what are some ideas on which problems to use as the basis for a polynomial-time reduction?

1 Answer

Suppose in a graph $G$ there is one pebble on each vertex except one vertex $v$ with $p(v) = 2$, then above pebbling problem has solution on $G$ iff $G$ has a Hamiltonian circuit. It’s easy to check if there is a Hamiltonian circuit, then there is a solution for pebbling on $G$. On the other hand, in any solution to the pebbling, we should start from vertex $v$. Suppose that we visit some vertex $u$ twice such that this $u$ is the first vertex which visited twice in $G$ by pebbling algorithm, then we have a loop which starts from $u$ and ends in $u$ and finally because $u$ is the first for making loop then we have $p(u) = 1$ so we cannot continue pebbling algorithm. Indeed if the algorithm has a solution then we have $u = v$ which means we found a Hamiltonian circuit which starts in $v$.

Choosing what to reduce From:
- Finding large subset?
- Finding small subset?
- Finding subset with specific properties?
- Labeling/classifying?
- Long sequence?
- Balancing/packing?
- 3 something?
- Give up?

Max Clique
Max Ind Set
Min Vertex Cover
SAT
Coloring
Hamiltonian something
(3) Partition
3 SAT/3 Color/3 Partition
3SAT/Circuit SAT
Some useful NP-hard problems. You are welcome to use any of these in your own NP-hardness proofs, except of course for the specific problem you are trying to prove NP-hard.

**CIRCUITSat:** Given a boolean circuit, are there any input values that make the circuit output True?

**3SAT:** Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?

**MAXINDEPENDENTSET:** Given an undirected graph $G$, what is the size of the largest subset of vertices in $G$ that have no edges among them?

**MAXCLIQUE:** Given an undirected graph $G$, what is the size of the largest complete subgraph of $G$?

**MINVERTEXCOVER:** Given an undirected graph $G$, what is the size of the smallest subset of vertices that touch every edge in $G$?

**MINSETCOVER:** Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set $S$, what is the size of the smallest subcollection whose union is $S$?

**MINHITTINGSET:** Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set $S$, what is the size of the smallest subset of $S$ that intersects every subset $S_i$?

**3COLOR:** Given an undirected graph $G$, can its vertices be colored with three colors, so that every edge touches vertices with two different colors?

**HAMILTONIANPATH:** Given graph $G$ (either directed or undirected), is there a path in $G$ that visits every vertex exactly once?

**HAMILTONIANCYCLE:** Given a graph $G$ (either directed or undirected), is there a cycle in $G$ that visits every vertex exactly once?

**TRAVELINGSALESMAN:** Given a graph $G$ (either directed or undirected) with weighted edges, what is the minimum total weight of any Hamiltonian path/cycle in $G$?

**LONGESTPATH:** Given a graph $G$ (either directed or undirected, possibly with weighted edges), what is the length of the longest simple path in $G$?

**STEINERTREE:** Given an undirected graph $G$ with some of the vertices marked, what is the minimum number of edges in a subtree of $G$ that contains every marked vertex?

**SUBSETSUM:** Given a set $X$ of positive integers and an integer $k$, does $X$ have a subset whose elements sum to $k$?

**PARTITION:** Given a set $X$ of positive integers, can $X$ be partitioned into two subsets with the same sum?

**3PARTITION:** Given a set $X$ of $3n$ positive integers, can $X$ be partitioned into $n$ three-element subsets, all with the same sum?

**INTEGERLINEARPREDICTING:** Given a matrix $A \in \mathbb{Z}^{n \times d}$ and two vectors $b \in \mathbb{Z}^n$ and $c \in \mathbb{Z}^d$, compute $\max \{ c \cdot x | Ax \leq b, x \geq 0, x \in \mathbb{Z}^d \}$.

**FEASIBLEILP:** Given a matrix $A \in \mathbb{Z}^{n \times d}$ and a vector $b \in \mathbb{Z}^n$, determine whether the set of feasible integer points $\max \{ x \in \mathbb{Z}^d | Ax \leq b, x \geq 0 \}$ is empty.

**DRAUGHTS:** Given an $n \times n$ international draughts configuration, what is the largest number of pieces that can (and therefore must) be captured in a single move?

**SUPERNARIOSBROTHERS:** Given an $n \times n$ Super Mario Brothers level, can Mario reach the castle?

**STEAMEDHAMS:** Aurora borealis? At this time of year, at this time of day, in this part of the country, localized entirely within your kitchen? May I see it?
Reduce 3SAT to HamCycle

variable gadgets

clause gadgets

\((a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b} \lor \bar{d})\)

\(a = b = d = \text{TRUE},\ c = \text{FALSE}\)
Classic Nintendo Games are (Computationally) Hard

Greg Aloupis*    Erik D. Demaine†    Alan Guo‡‡    Giovanni Viglietta§

February 10, 2015
Solving the Rubik’s Cube Optimally is NP-complete

Erik D. Demaine
MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA
edemaine@mit.edu

Sarah Eisenstat
MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA

Mikhail Rudoy¹
MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA
mrudoy@gmail.com
draughts
- Flying kings!
- Captured pieces stay to end of move
- Forced capture max # pieces
\[ O(n^3) \text{ edges} \]
\[ \rightarrow O(n^4) \text{ crossing gadgets} \]
So need \( \geq n^3 \) black pieces in each vertex gadget use \( \Theta(n^4) \)

So width = \( \Theta(n^5) \), area = \( \Theta(n^{10}) \)

\text{polynomial!}