NPC: Proving and Coping

CS 350 Guest Lecture

Bart Massey <bart@cs.pdx.edu>
PSU CS Department
February 21, 2001

Overview

- Some Example Reductions
- Polytime Approximation Schema
- Search
- The Blocks World

Some Example Reductions

Why?

- Get used to reductions
- Graduated examples
- Different classes of problems

The Plan

```
    csat
   /
sat-3sat-clique-vertex cover-subset sum
        \
         ham cycle-tsp
```

SAT

Problem: Given formula in boolean vars

$$\neg(A \wedge B) \wedge (\neg A \vee B)$$

find satisfying assignment

Is NPC (Cook's Theorem, from last time)

From SAT To Circuit SAT

Boolean Circuit SAT is NPC:

- Construct binary tree representation of formula
- Is Boolean Circuit outputting 1 iff SAT

From SAT To 3-SAT

3-CNF-SAT (aka 3-SAT) NPC

- Construct binary tree
- Build formula (each clause 3 vars)
- Build DNF for each clause negation
- DeMorganize into CNF for clause

From 3-SAT To CLIQUE

Note: Text says "clique = K", not "clique ¿= K"

- Idea: encode formula as graph
  - 1 node/literal (i.e 3/clause)
  - edges connect all literals in different clauses except negated
  - k clauses satisfiable iff k-clique
    - if: true literals form clique
    - only if: literals in clique all true

From CLIQUE To VERTEX-COVER

- Find vertices s.t. all edges hit
- Take complement G' of G
- G' has k-clique iff G has $|V| - k$ -cover

From VERTEX-COVER To SUBSET-SUM

See text. :-)

From 3-SAT To HAM-CYCLE

Hamiltonian Cycle: path through graph hitting each node exactly once

Take 3-SAT to HAM-CYCLE? Build fancy graphs with structures corresponding to clauses, literals, vars, etc!

Classic scary reduction

Gadget A

Gadget B

The Construction

Use A for var/clause constraint and B for clause sat constraint

From HAM-CYCLE To TSP

Easy final redn: TSP is shortest weighted cycle. Can reduce HAM-CYCLE to TSP with weights 0, 1 for edges/non-edges of graph. TSP has 0 cycle iff graph has Ham cycle

Polytime Approximation Schema

One way to cope with NPC optimization problem: find polytime "close" approximation

- Close = good enough for purpose (satisficing)
- Polytime = stays close but computable in $O(n^k)$

Approximation Bounds

Ratio bound:

$$C/C^* \leq \rho(n)$$

Relative bound:

$$\frac{C - C^*}{C^*} \leq \epsilon(n)$$

Fixed bound: independent of n

PTAS: Given ϵ, can find AS with relative bound ϵ running in polytime

Fully Polytime Approximation Schema

Like PTAS, but also with time polynomial in $1/\epsilon$

Can make relative error as small as desired for n as large as desired without losing polytime

A Ratio PTAS For Vertex Cover

Algorithm for Vertex Cover which is

- polytime
- within factor of 2 of optimal

Pick edge, add both ends to cover, delete all edges impinging on either end. Stop when no edges

Idea: each edge must have at least one vertex in optimal cover, and we only put in two per

A Ratio PTAS For Geometric TSP

Again, within factor of 2

Construct minimal spanning tree for problem. Now delete all but first visit to a vertex.

- Shortens (because of triangle inequality)
- Makes tour

Note that can still improve: e.g. “uncrossing”

No PTAS For General TSP

Idea: could use PTAS to solve HAM-CYCLE in polytime. Thus, unless $P = NP$, PTAS does not exist.

Search

What if

- There is no PTAS?
- You can't find one?
- Not an optimization problem?
- Polynomials too large?

but solution still needed to instances?

Answer: "search" = educated guessing

NPC Problems, Certificates, and Decisions

NPC is class of "guess + check". Guess what? Certificate = solution object

Can guess certificate piece-at-a-time? Take one big guess to many little guesses

Cover all possible certificates *systematically* in "search space"

Organizing Brute Force DFS

- Start with no guesses
- Make one at a time
- When all have been made, at leaf of search tree
- No solution? *Backtrack* to try last guess differently

Result: Depth First Search (systematic, complete)

Heuristics

- Probably want to try likely guesses first
- Heuristic is rule-of-thumb toward guessing right
- Conventionally, search tree is organized with best guesses on left
- Which part of certificate (“variable”) first?

DFS Considered Harmful

Search tree is too big to actually search! What portion does DFS search?

- Closely related solutions
- Flawed by early mistakes

Other search algorithms exist which trade systematicity for efficiency

Stochastic Search

Give up on completeness altogether? (Were not getting it anyhow)

"Simulated Annealing" is example of *stochastic search* method: try guided random changes to broken certificate until it is *repaired*

Can be highly efficient in practice!

Why Search?

What instances of NPC problem are interesting? Usually

- Bounded in size (but large)
- Not polytime subclass
- Mostly solvable

Good search gets big jump on size of interesting instance. In practice, "interesting instances" similar enough that given technique either works or not on all

The Blocks World

One of earliest AI problems. Set of (uniquely) labeled blocks on infinite table.

Blocks World Planning

The BWP problem: given initial, goal block configurations, give sequence of TOS moves from initial to goal

BWP In NP?

Is BWP in NP?

- Not obvious: is Hanoi in NP?
- But obvious: child's technique works
- Nonetheless, general-purpose planning doesn't work so well

Optimal BWP Is NPC

Is finding *shortest* plan easy? No, NPC (reduction from HITTING SET, Gupta and Nau 1990)

Seems easy, but problem is "deadlock": need to move block to table, but which?

No FPTAS For OBWP

Reduction shows there is no FPTAS for OBWP. But ratio factor-of-two scheme still holds! (Do stack-unstack preserving correct subtowers)

Efficient Heuristics For OBWP

Note that some things easy:

- Block can move to final position? Move it
- Block must move twice? Move it
- Leaves deadlock case: can use heuristics to try to break

Result: fast OBWP solver