EECS 336: Lecture 9: Introduction to Algorithms

P vs. NP: indep set, 3-sat, TSP

Reading: 8.0-8.3

`"guide to reductions"`

Last Time:

- max flow alg / ford-fulkerson
- duality: max flow = min cut

Today:

- reductions (cont)
- tractability and intractability
- decision problems
- 3-SAT $\leq_P$ INDEP-SET

Reduction Illustrated

Problems	Bipartite Matching	Network Flow
Instance	$x = (A, B, E)$	$y^x = (V^x, E^x, c^x, s^x, t^x)$
Solution	M	f^x

Summary of Reduction

Def: <u>Y reduces to X</u> in polynomial time (notation: $Y \leq_P X$) if any instance of Y can be solved in a polynomial number of computational steps and a polynomial number of calls to <u>black-box</u> that solves instances of X.

Note: to prove correctness of general reduction, must show that correctness (e.g., optimality) of algorithm for X implies correctness of algorithm for Y.

Def: <u>one-call reduction</u> maps instance of Y to instance of X, solution of Y to solution of X. (also called a Karp reduction)

Note: a one-call reduction gives two algorithms:

I. contruction of X^Y instance from Y instance.

II. construction of Y solution from X^Y solution (with same value.)

Note: the proof of correctness of a one-call reduction gives additional algorithm:

III. construction of X^Y solution from Y solution (with same value.)

Note: Only need to consider X^Y instance not general X instance.

Note: If solution not needed then reduction is Step I and proof is Steps II and II.

Theorem: reduction from "I and II" is correct if I, II, and III are correct.

Proof:

- for instance y of Y, let instance of x^y of X^Y be outcome of I.
- II correct $\Rightarrow$ OPT$(y) \geq$ OPT(x^y).
- III correct $\Rightarrow$ OPT$(x^y) \geq$ OPT(y).

$\Rightarrow$ OPT$(y) =$ OPT(x^y).

$\Rightarrow$ output of reduction has value OPT(y).

Decision Problems

"problems with yes/no answer"

Def: A decision problem asks "does a feasible solution exist?"

Example: network flow in (V, E, c, s, t) with value at least θ.

Example: perfect matching in a bipartite graph (A, B, E).

Note: objective values for decision problem is 1 for "yes" and 0 for "no".

Note: II and III only need to check "yes" instances.

Theorem: perfect matching reduces to network flow decision problem.

Note: Can convert optimization problem to decision problem

Def: the decision problem X_d for optimization problem X has input (x, θ) = "does instance x of X have a feasible solution with value at most (or at least) θ?"

Tractability and Intractability

Consequences of $Y \leq_p X$:

1. if X can be solved in polynomial time then so can Y.

   Example: $X =$ network-flow; $Y =$ bipartite matching.

2. if Y cannot be solved in polynomial time then neither can X.

Reductions for Intractability

"reduce known hard problem Y to problem X to show that X is hard"

Problem Y: 3-SAT

input: blooen formula $f(\mathbf{z}) = \bigwedge_{j=1}^{m} (l_{j1} \vee l_{j2} \vee l_{j3})$

- literal l_{jk} is variable "z_i" or negation "$\bar{z}_i$"
- "and of ors"
- e.g., $f(\mathbf{z}) = (z_1 \vee \bar{z}_2 \vee z_3) \wedge (z_2 \vee \bar{z}_5 \vee z_6) \wedge ...$

output:

- "Yes" if assigment $\mathbf{z}$ with $f(\mathbf{z}) = T$ exists

  e.g., $\mathbf{z} = (T, T, F, T, F, ...)$
- "No" otherwise.

Problem X: INDEP-SET

input: $G = (V, E), k$

ouput: "yes" if $\exists S \subset V$

- satisfying $\forall v \in S, (u, v) \notin E$
- $|S| \geq \theta$

Reduction

Lemma: 3-SAT $\leq_p$ INDEP-SET

Part 1: forward instance construction

convert 3-SAT instance f into INDEP-SET instance (V^f, E^f, θ^f).

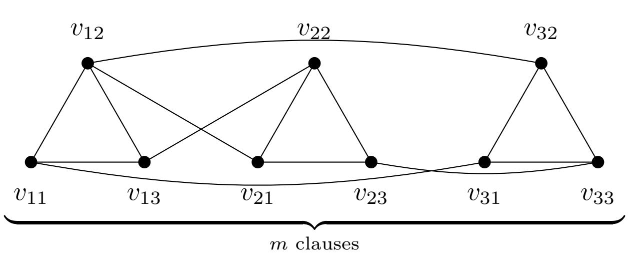
- goal: "at least one true literal per clause" $\Leftrightarrow$ "independent set of size at least θ"
- literal $l_{ij} \Rightarrow$ vertices $v_{ij} \in V^f$
- "all clauses true" $\Rightarrow \theta^f = m$
- "literal conflicts" $\Rightarrow$ conflict edges.

  $\forall i$: $l_{jk} =$ "z_i" and $l_{j'k'} =$ "$\bar{z}_i$" $\Rightarrow (v_{jk}, v_{j'k'}) \in E^f$
- "one representative per clause" $\Rightarrow$ clause edges.

  $\forall j$: $(v_{j1}, v_{j2}),\ (v_{j2}, v_{j3}),\ (v_{j3}, v_{j1}) \in E^f$

Example:

$f(\mathbf{z}) = (z_1 \vee z_2 \vee z_3) \wedge (\bar{z}_2 \vee \bar{z}_3 \vee \bar{z}_4) \wedge (\bar{z}_1 \vee \bar{z}_2 \vee z_4)$



Runtime Analysis: linear time (one vertex per literal.)

Part II: reverse certificate construction

construct assignment $\mathbf{z}$ from S^f

(if (V^f, E^f) has indep. set S^f size $\geq \theta^f = m$ then f is satisfiable.)

1. For each z_r:

   (a) if exists vertex in S^f labeled by "z_r"

   set $z_r = T$

   (b) else

   set $z_r = F$

Claim: if vertex in S is labeled by "$\bar{z}_r$" then no vertices in S are labeled by "z_r" and z_r is set to False.

(because of conflict edge between vertex labeled "$\bar{z}_r$" and all vertices labeeleed "z_r".)

Claim: S^f independent and $|S^f| \geq m \Rightarrow f(\mathbf{z}) = T$:

- S has $|S| = m$

  $\Rightarrow S$ has one vertex per clause.
- for clause i and $v_{ij} inS$:

  if l_{ij} not negated, then z_i is true (by construction)

  if l_{ij} is negated then z_i is false (by claim)
- So $f(\mathbf{z}) = T$.

Part III: forward certificate construction

construct independent set S^f from $\mathbf{z}$

(if f is satisfiable then (V^f, E^f) has indep set size $\geq m = \theta^f$.)

1. let S' be nodes in (V^f, E^f) corrpesonding to true literals.
2. if more than one vertex in S' in same triangle drop all but one.

   $\Rightarrow S^f$.

Claim: $\mathbf{z}$ satisfies $f(\mathbf{z}) \Rightarrow S^f$ independent and $|S^f| \geq m$

- all clauses have true literal

  $\Rightarrow |S'| \geq m$ and $|S| = m$
- for all $u, v \in S$,
  - u & v not in same triangle.
  - l_u and l_v both true

    $\Rightarrow$ must not conflict

    $\Rightarrow$ no (l_u, l_v) edge in (V^f, E^f).
  - so S^f is independent.