

Lecture # 24

Lecturer: Debmalya Panigrahi

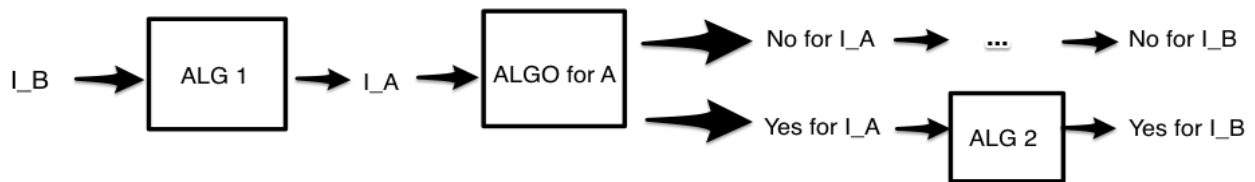
Scribe: Ang Li

1 Overview

This lecture covers some common examples of NP-complete problems, polynomial-time reductions between some of them, and finally an introduction to approximation algorithm.

2 Polynomial Reduction

Given an instance of Problem B I_B , if there is a polynomial-time Algorithm 1 that we can use to convert the problem to an instance of Problem A I_A , such that a "Yes" answer for I_A means a "Yes" answer for I_B and a "No" answer for I_A means a "No" answer for I_B through a polynomial-time transforming Algorithm 2, we say that Problem B can be reduced to Problem A and that Problem A is at least as hard as Problem B.



Note that both ALG 1 and ALG 2 have to be polynomial-time.

3 Decision Problems & Optimization Problems

Decision problems have either yes or no answer, while optimization problems outputs a value as the answer.

Example 1. The Traveling Salesman Problem.

Optimization version: find the smallest-length tour in a given graph.

Decision version: is there a tour in the graph of length $\leq k$.

Lemma 1. *Polynomial-time algorithm for Decision(TSP) is equivalent to polynomial-time algorithm for Optimization(TSP).*

Proof.

Converting from Optimization(TSP) to Decision(TSP): run it once and compare the answer to value k .

Converting from Decision(TSP) to Optimization(TSP): naive solution is to increment k by 1 every time and check whether it's "Yes" or "No" using the algorithm for Decision(TSP). However, this could well be a pseudo-polynomial algorithm since it's dependent on the size of the answer k . Another method is to double k every iteration initially and then do binary search once a boundary is found; this algorithm will be weakly polynomial. □

4 Reduction Examples

4.1 Vertex Cover & Independent Set

Definition 1. *Vertex Cover:* Given a graph $G = (V, E)$, find $S \subseteq V$ such that for every $e = (u, v) \in E$, $|S \cap \{u, v\}| \geq 1$, and $|S|$ is minimized.

Definition 2. *Independent Set:* Given a graph $G = (V, E)$, find $S \subseteq V$ such that for every $e = (u, v) \in E$, $|S \cap \{u, v\}| \leq 1$, and $|S|$ is maximized.

4.1.1 VC \leq IS (Reduction from VC to IS)

Given an instance of vertex cover, i.e. in a graph $G = (V, E)$ is there a vertex cover of size $\leq k$. This problem is equivalent to (by ALG 1) to this instance of IS: in a graph $G = (V, E)$ is there an independent set of size $\geq n - k$.

Proof.

No case:

We want to show that if there is no independent set of size $\geq n - k$, then there is no vertex cover of size $\leq k$, which is equivalent to proving that if there is a vertex cover of size $\leq k$, then there is an independent set of size $\geq n - k$. The proof is easy: if S is a vertex cover and $|S| \leq k$, then $V \setminus S$ is an independent set and $|V \setminus S| \geq n - k$.

Yes case:

Suppose S is an independent set of size $\geq n - k$, then a vertex cover of size $\leq k$ is simply $V \setminus S$ (ALG 2). \square

4.2 Independent Set & Clique

Definition 3. *Clique problem:* A clique is a subset of vertices such that every two share an edge. Given a graph $G = (V, E)$, find the clique of the largest size.

4.2.1 Clique \leq, \geq IS

Given a clique problem, i.e. given $G = (V, E)$ and k , it can be transformed to an independent set problem (ALG 1) with $G^c = (V, E^c)$ and k , where $E^c = \{(u, v) : (u, v) \notin E\}$.

Proof.

No case:

If S is the set of vertices that form a clique in the original graph G , then S in the transformed graph G^c is an independent set by the definitions of clique and independent set.

Yes case:

If S is the set of vertices that form an independent set in G^c , then S in the original graph G is a clique by the definitions of clique and independent set. \square

4.3 3-SAT \leq Independent Set

Given a 3-SAT problem: $(x_1 \vee x_2 \vee \bar{x}_3)$ AND $(\bar{x}_2 \vee x_3 \vee x_4)$, is there a way of assigning the boolean variables such that the sentence evaluates to true. This can be transformed to an independent set problem in the following way (ALG 1). To generate graph G , form a triangular clique for every clause in the 3-SAT problem, with the three vertices corresponding to the variables in the clause, then add edges between opposite literals in the entire graph. k will be equal to the number of clauses.

Proof.

No case:

If we have a solution for the 3-SAT problem, we can pick the vertices corresponding to the true literals in the satisfying assignment to form the independent set. If there're multiple literals set to true in a clause, pick any one and only one corresponding vertex from that triangle. This set of vertices is an independent set of size k in the constructed graph because only one vertex from each triangular clique is picked, and that no opposite literals are picked (such an assignment is impossible).

Yes case:

If we have a solution for the constructed independent set problem, we can set the variables that correspond to the chosen vertices to true. This creates a satisfying assignment because for every triangular clique, there is one and exactly one vertex chosen in the independent set for its size to be at least k . Furthermore, no variable can be both true and false since there is an edge between their vertices in the constructed graph. \square

5 Approximation Algorithms

5.1 A greedy algorithm for Vertex Cover

Start with S being an empty set while E is not empty

pick $(u, v) \in E$

add both u and v to S

remove all edges incident on u or v from E

5.2 Approximation ratio

Definition 4. For minimization problems, approximation ratio = $\max_{\text{over all instances}} \frac{\text{Algo solution}}{\text{Opt solution}}$

Approximation algorithms try to minimize the approximation ratio while keeping the algorithm in polynomial running time.

5.3 Analysis for the greedy Vertex Cover algorithm

From the steps in the algorithm, $\text{Size of ALGO solution} = \sum_{e \in \text{ALGO}} 2$. Observe that $\{e \in \text{ALGO}\}$ do not share any endpoint. This is because all edges incident on the included vertices are immediately removed.

This gives $\text{size of OPT solution} \geq \sum_{e \in \text{ALGO}} 1$ for the solution to be a vertex cover by definition.

As a result, approximation ratio = $\frac{\text{ALGO}}{\text{OPT}} \leq 2$