Lecturer: Hung Q. Ngo
Scribe: Dazhen Pan

Lecture 1: Approximation Algorithms, Approximation Ratios, Gap Problems

1 Introduction

To date, thousands of natural optimization problems have been shown to be NP-hard [6, 13]. Designing approximation algorithms [4, 17, 21] has become a standard path to attack these problems. For some problem, however, it is even NP-hard to approximate the optimal solution to within a certain ratio. The TRAVELING SALESMAN PROBLEM (TSP), for instance, has no approximation algorithm, since finding a feasible solution (the HAMILTONIAN CIRCUIT problem) is already NP-hard.

Until 1990, few inapproximability results were known. The reader is referred to recent surveys by Feige [10] and Trevisan [20] for good discussions on this point and related history. To prove a typical inapproximability result such as MAX-CLIQUE is not approximable to within some ratio r (unless P=NP), a natural direction is to find a reduction from some NP-complete problem, say 3SAT, to MAX-CLIQUE which satisfies the following properties:

- given a 3CNF formula ϕ, the reduction constructs in poly-time a graph G_ϕ
- there is some poly-time computable *threshold* t such that, if ϕ is satisfiable, then G_ϕ has a clique of size at least t, and if ϕ is not satisfiable, then G_ϕ does not have any clique of size t/r or more.

If MAX-CLIQUE is r-approximable, then one can use this r-approximation algorithm, along with the reduction above, to decide if a 3CNF formula ϕ is satisfiable. The strategy is to run the algorithm on G_ϕ. If the answer is t/r or more, then ϕ is satisfiable, else ϕ is not.

Techniques for proving NP-hardness seem inadequate for this kind of *gap-producing* reductions. Intuitively, the reason is that non-deterministic Turing Machines are sensitive to small changes: the accepting computations and rejecting computations are not very far from one another (no gap). In 1990, the landmark work by Feige, Goldwasser, Lovász, Safra, and Szegedy [11] connects probabilistic proof systems and inapproximability of NP-hard problems. This has become known as **the** PCP connection. A year later, the PCP theorem - a very strong characterization of NP - was proved with the works of Arora and Safra [3], Arora, Lund, Motwani, Sudan, and Szegedy [2]. A plethora of inapproximability results using the PCP connection follow, some of them are even optimal [1, 7, 9, 12, 14–16, 18].

In this seminar, we will focus on the PCP connection and move spirally wider around. We follow the line of presentation by Bellare, Goldreich, and Sudan [5]. The paper is very instructive! After that, we shall discuss papers by Håstad [15, 16], Raz [19], and related techniques such as Fourier analysis.

2 Approximation Algorithms and Approximation Ratios

We begin with several definitions laying out a framework to analyze inapproximability results.

Definition 2.1 (Promise problem). A *promise problem* [8] consists of a pair of disjoint finite sets (A, B). An input x to the problem is promised to be in $A \cup B$. The problem is to decide if x is in A or B, and answer YES or NO, respectively. The normal membership decision problem for a language L is the promise problem $(L, \bar{L})$.

Definition 2.2 (Optimization problem). An *optimization problem* Φ is a tuple

$$\Phi = (S, g, \| \cdot \|, | \cdot \|^*, \textsc{opt}),$$

in which

- S is a function which, on input instance w, gives the set $S(w)$ of all feasible solutions to w;
- g is the objective function, i.e. for any $y \in S(w)$, $g(y, w)$ is the cost of the solution y;
- $\textsc{opt} \in \{\max, \min\}$ indicates if Φ is a maximization or a minimization problem;
- $\| \cdot \|$ is a polynomially bounded and polynomial time computable function on w (also called *admissible* function) which gives us a more convenient measure on the size of the instance w; (For example, if w is a graph, then $\|w\|$ is often the number of vertices of the graph.)
- $\| \cdot \|^*$ is a polynomial time computable function on w, which will be used to normalize the objective value so that the objective value is between 0 and 1. It shall be clear later why we need this normalization.

As an example, for $\Phi = \textsc{Max-Clique}$ we have: w is some graph G, $S(G)$ is the set of all cliques of G, g gives size of a clique, $\textsc{opt} = \max$, $\|G\|$ and $\|G\|^*|$ are equal to the number of vertices of G.

Definition 2.3 (Approximation algorithm). An *approximation algorithm* A is a polynomial time algorithm that computes some value close to the optimal solution of an optimization problem Φ. Let $A(w)$ denote the value that A returns.

Definition 2.4 (Approximation ratio). An approximation algorithm A for Φ has **approximation ratio** μ_A (a function of $\|w\|$) if, for all instances w of Φ,

$$\frac{\Phi(w)}{\mu_A} \leq A(w) \leq \Phi(w), \quad \text{when } \textsc{opt} = \max,$$

and

$$\Phi(w) \leq A(w) \leq \mu_A \Phi(w), \quad \text{when } \textsc{opt} = min.$$

Here,

$$\Phi(w) = \begin{cases} \max\limits_{y \in S(w)} g(y, w), & \text{if } \textsc{opt} = \max \\ \min\limits_{y \in S(w)} g(y, w), & \text{if } \textsc{opt} = \min. \end{cases}$$

We use $\bar{\Phi}(w)$ to denote the normalized objective value:

$$\bar{\Phi}(w) = \frac{\Phi(w)}{\|w\|^*}.$$

3 Gap Problems

Consider an optimization problem $\Phi = (S, g, \| \cdot \|, | \cdot \|^*, \textsc{opt})$. Let $c, s : \mathbb{N}^+ \to [0, 1]$ be two admissible functions of the norm $\| \cdot \|$ such that $0 < s(n) < c(n) \leq 1, \forall n$. Define the *gap version* $\textsc{Gap}\text{-}\Phi_{c,s}$ of Φ as follows. $\textsc{Gap}\text{-}\Phi_{c,s}$ is a promise problem (Y, N), where

$$\left. \begin{array}{l} Y = \{w : \bar{\Phi}(w) \geq c(\|w\|)\} \\ N = \{w : \bar{\Phi}(w) < s(\|w\|)\} \end{array} \right\} \text{when } \textsc{opt} = \max,$$

and

$$\left.\begin{array}{l} Y = \{w : \bar{\Phi}(w) \leq s(\|w\|)\} \\ N = \{w : \bar{\Phi}(w) < c(\|w\|)\} \end{array}\right\}, \text{when } \textsc{opt} = \min.$$

Let $\leq_D^K$ denote the *deterministic Karp reduction*. We write NP $\leq_D^K \Pi$ if any problem in NP is deterministically Karp-reducible to the problem Π. In particular, NP $\leq_D^K \Pi$ if some NP-complete problem is reducible to Π.

Proposition 3.1. *If NP $\leq_D^K$ $\textsc{Gap}$-$\Phi_{c,s}$, then it is not possible to approximate Φ to within $\frac{c}{s}$, unless P=NP.*

Proof. The idea is that, if there exists a $\frac{c}{r}$-approximation algorithm A for Φ, then there is an polynomial time algorithm B that can decide membership of any NP-complete problem, say *SAT*.

For the convenience of presentation, we will use c, s to denote $c(\|w\|), s(\|w\|)$. Assume $\textsc{opt} = \max$ (the other case is similar). Let

$$\bar{A}(w) = \frac{A(w)}{\|w\|^*}$$

Because A approximates Φ to within $\frac{c}{s}$, we have

$$\frac{\Phi(w)}{\frac{c}{s}} \leq A(w) \leq \Phi(w),$$

which implies

$$\frac{\bar{\Phi}(w)}{\frac{c}{s}} \leq \bar{A}(w) \leq \bar{\Phi}(w). \tag{1}$$

Given a boolean formula ϕ, we first use the deterministic Karp-reduction to get an instance w of $\textsc{Gap}$-$\Phi_{c,s}$. By definition, ϕ is satisfiable iff $\bar{\Phi}(w) \geq c$, and ϕ is not satisfiable iff $\bar{\Phi}(w) < s$.

To decide if ϕ is satisfiable, we run A on w. If $\bar{A}(w) \geq s$, then $\bar{\Phi}(w) \geq \bar{A}(w) \geq s$, and thus $\bar{\Phi} \geq c$, which implies ϕ is satisfiable. On the other hand, if $\bar{A}(w) < s$, then $\bar{\Phi}(w) \leq \frac{c}{s}\bar{A}(w) < c$, and thus $\bar{\Phi} < s$, which implies ϕ is not satisfiable. ☐

This proposition formalizes the idea of gap-producing reductions we discussed in Section 1. The key is obviously the assumed existence of such a reduction. Non-deterministic Turing machines seem inadequate for this task. The PCP characterization of NP gives us a much better handle on this reduction. We will discuss PCP in the next lecture.

References

[1] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, J. Comput. System Sci., 54 (1997), pp. 317–331. 34th Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1993).

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555. Prelim. version in FOCS'92.

[3] S. Arora and S. Safra, *Probabilistic checking of proofs: a new characterization of NP*, J. ACM, 45 (1998), pp. 70–122. Prelim. version in FOCS'92.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and approximation*, Springer-Verlag, Berlin, 1999. Combinatorial optimization problems and their approximability properties, With 1 CD-ROM (Windows and UNIX).

[5] M. Bellare, O. Goldreich, and M. Sudan, *Free bits, PCPs, and nonapproximability—towards tight results*, SIAM J. Comput., 27 (1998), pp. 804–915 (electronic).

[6] P. Crescenzi and V. K. (eds.), *A compendium of NP-optimization problems*. http://www.nada.kth.se/

[7] I. DINUR AND S. SAFRA, *On the hardness of approximating label-cover*, Inform. Process. Lett., 89 (2004), pp. 247–254.

[8] S. EVEN, A. L. SELMAN, AND Y. YACOBI, *The complexity of promise problems with applications to public-key cryptography*, Inform. and Control, 61 (1984), pp. 159–173.

[9] U. FEIGE, *A threshold of* $\ln n$ *for approximating set cover (preliminary version)*, in Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996), New York, 1996, ACM, pp. 314–318.

[10] ———, *Approximation thresholds for combinatorial optimization problems*, in Proceedings of the International Congress of Mathematicians, Vol. III (Beijing, 2002), Beijing, 2002, Higher Ed. Press, pp. 649–658.

[11] U. FEIGE, S. GOLDWASSER, L. LOVÁSZ, S. SAFRA, AND M. SZEGEDY, *Interactive proofs and the hardness of approximating cliques*, J. ACM, 43 (1996), pp. 268–292. Prelim. version in FOCS'91.

[12] U. FEIGE AND J. KILIAN, *Zero knowledge and the chromatic number*, J. Comput. System Sci., 57 (1998), pp. 187–199. Complexity 96—The Eleventh Annual IEEE Conference on Computational Complexity (Philadelphia, PA).

[13] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[14] V. GURUSWAMI, J. HÅSTAD, AND M. SUDAN, *Hardness of approximate hypergraph coloring*, SIAM J. Comput., 31 (2002), pp. 1663–1686 (electronic).

[15] J. HÅSTAD, *Clique is hard to approximate within* $n^{1-\epsilon}$, Acta Math., 182 (1999), pp. 105–142.

[16] ———, *Some optimal inapproximability results*, in STOC '97 (El Paso, TX), ACM, New York, 1999, pp. 1–10 (electronic).

[17] D. S. HOCHBAUM, ed., *Approximation Algorithms for NP Hard Problems*, PWS Publishing Company, Boston, MA, 1997.

[18] C. LUND AND M. YANNAKAKIS, *On the hardness of approximating minimization problems*, J. Assoc. Comput. Mach., 41 (1994), pp. 960–981.

[19] R. RAZ, *A parallel repetition theorem*, SIAM J. Comput., 27 (1998), pp. 763–803 (electronic).

[20] L. TREVISAN, *Inapproximability of combinatorial optimization problems*, Tech. Rep. 65, The Electronic Colloquium in Computational Complexity, 2004.

[21] V. V. VAZIRANI, *Approximation algorithms*, Springer-Verlag, Berlin, 2001.