E0 249: Approximation Algorithms

Naman Mishra

January 2024

Contents

Lecture 01: Mon 08 Jan '24

0 The Course

Course website: here
MS Teams: 091kg9h
Instructors: Prof. Arindam Khan and Prof. Anand Louis
TAs: Aditya Subramaniam (January)
Lecture Hours: MW 1400–1530 at CSA 112
Office Hours: Just email.

0.1 Grading

(30%) Homework

(20%) Project. Research papers from top conferences (STOC, FOCS, SODA, ICALP, SoCG). Around 10% on a report. 10% on a presentation (half an hour).

(20%) Midterm

(30%) Final

0.2 Texts

Primarily *The Design of Approximation Algorithms* by David Williamson and David Shmoys, Cambridge University Press, 2011. Available online for free.
Alternatively, *Approximation Algorithms* by Vijay Vazirani, Springer-Verlag, 2001.
For specific topics, Hochbaum, Sariel, etc.

1 Optimization Problems

Find the *best* solution from a set of *feasible* solutions. Richard Karp introduced the concept of NP-complete problems. Unless P = NP, the optimization version of the problems admit no algorithms that simultaneously (1) find optimal solution (2) in polynomial time (3) for all instances.

1.1 Optimization Version of Common NP-Complete Problems

Problem	Optimization Version
3-SAT	Max 3-SAT
Independent Set	Max Independent Set
Vertex Cover	Min Vertex Cover
Set Cover	Min Set Cover
Hitting Set	Min Hitting Set
Clique	Max Clique

1.2 Approximation Algorithms

Task: Solve NP-hard optimization problems A, but no efficient algorithm exists (unless P = NP).

Definition 1.1. Let Π be an optimization problem and let I be an instance of Π. Then $\mathrm{OPT}_\Pi(I)$ is the value of the optimal solution.

There might be several optimal solutions, but they all have the same value.

Definition 1.2. Let $\alpha \geq 1$. A is an α-approximation algorithm for a minimization problem Π if for every instance I of Π,

$$A(I) \leq \alpha \, \mathrm{OPT}_{\Pi}(I)$$

where $A(I)$ is the value of the solution that A returns for I.

Typically, α takes values like 1.5, 2, $O(1)$, $O(\log n)$, etc.
Usually we omit Π and I in $\mathrm{OPT}_{\Pi}(I)$.

For a maximization problem, $A(I) \geq \frac{1}{\alpha} \mathrm{OPT}_{\Pi}(I)$. (Sometimes in literature, $\alpha \leq 1$ is used for maximization problems.)

This is also called *absolute approximation*.

NP-hard problems are very similar to each other in terms of decidability, but can be very different in terms of approximability. For some problems, it is NP-hard to obtain any approximation (TSP) but for some (Knapsack) we can get $(1+\varepsilon)$-approximation in polynomial$(n, \frac{1}{\varepsilon})$ time.

Definition 1.3. A_ε is a polynomial time approximation scheme (PTAS) for a minimization problem Π if

$$A_\varepsilon(I) \leq (1+\varepsilon) \, \mathrm{OPT}(I)$$

and for every fixed $\varepsilon > 0$, the running time of A_ε is polynomial in n.

Definition 1.4 (EPTAS)**.** Efficient PTAS. $(1+\varepsilon)$-approximation in runtime $f(1/\varepsilon)n^{O(1)}$, where the exponent of n is independent of ε.

Definition 1.5 (FPTAS)**.** Fully polynomial time approximation scheme. $(1+\varepsilon)$-approximation in runtime polynomial in both n and $\frac{1}{\varepsilon}$.

Definition 1.6 (QPTAS)**.** Quasi-polynomial time approximation scheme. $(1+\varepsilon)$-approximation in quasi-polynomial time in n.

A *quasi-polynomial* is between a polynomial and an exponential.

Definition 1.7 (PPTAS)**.** Pseudo-polynomial time approximation scheme. $(1+\varepsilon)$-approximation in pseudo-polynomial time in n.

For example, $(nB)^{O(1)}$, where B is the biggest numeric data.

1.3 Asymptotic Approximation

Definition 1.8. The asymptotic approximation ratio ρ_A^∞ of an algorithm A is

$$\lim_{n\to\infty} \rho_A^n, \quad \text{where} \quad \rho_A^n = \sup_{I\in\mathcal{I}} \left\{ \frac{A(I)}{\mathrm{OPT}(I)} \mid \mathrm{OPT}(I) = n \right\}$$