Approximability August 25th, 2016
UT Austin CS395T

Syllabus

1 Introduction

This class is about approximation algorithms for NP-hard optimization problems and their limitations. The class meets on Tuesdays and Thursdays at 11-12:30 at GDC 4.302.

Instructor:	Dana Moshkovitz GDC 4.432 `danama@cs.utexas.edu` Office hours: coordinate by email
TA:	Ridwan Syed `ridwan@cs.utexas.edu` Office hours: Wednesday and Friday, 4-5, TA Desk 2
Website:	`http://www.cs.utexas.edu/~danama/courses/approximability/index.html`

2 Canvas

We will use Canvas for announcements, grading and discussions.

3 Textbooks

There are several textbooks that cover some of the material:

1. *David Williamson and David Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.* Covers combinatorial approximation algorithms, applications of linear programming and semidefinite programming to approximation algorithms, as well as hardness of approximation based on the PCP Theorem.

2. *Vijay Vazirani, Approximation Algorithms, Springer-Verlag, 2001.* Covers similar material to the previous book.

3. *Sanjeev Arora and Boaz Barak, Computational Complexity – A Modern Approach, Cambridge University Press, 2009.* Covers the Hadamard based construction of PCP, the combinatorial proof of the PCP Theorem, Fourier analysis. A draft of the book is available online free of charge, `http://www.cs.princeton.edu/theory/index.php/Compbook/Draft`, but the latest edition contains corrections and improvements.

4. *Ryan O'Donnell, Analysis of Boolean Functions, Cambridge University Press, 2014.* Covers Fourier analysis.

There are many lecture notes online that cover topics from the class. For approximation algorithms check out the webpages of: Anupam Gupta and Ryan O'Donnell, Chandra Chekuri, Shuchi Chawla and others. For hardness of approximation check out the webpages of: Dana Moshkovitz, Ryan O'Donnell and Venkat Guruswami, Subhash Khot.

4 Prerequisites

Undergraduate classes in algorithms and complexity theory.

5 Syllabus

- Basic combinatorial approximation algorithms
- Linear programming (LP) and approximation algorithms
- Semidefinite programming (SDP), Max-Cut approximation
- Hierarchies of linear and semidefinite programming and their limitations
- Hardness of Approximation, Multi-prover games and probabilistic checking of proofs (PCP)
- Linearity testing, Hadamard-based PCP, coding theory, Fourier analysis
- Sum-Check and PCP with polylog queries
- Composition, PCP of proximity, robust PCP, decoding PCP
- Gap amplification (powering)
- Parallel repetition
- The long code and optimal inapproximability results
- The Unique Games Conjecture, Dictator tests and integrality gaps

6 Workshops

Some of the meetings will be regular lectures, and some will be "workshops" in which the students give mini-lectures. Workshops will give students ample opportunity to give lectures and receive feedback. The workshops will be held on Tuesdays. On the Thursday before a workshop we will publish a schedule with the topics of the mini-lectures. Usually there will be problems to solve and a mini-lecture will present a solution, but sometimes mini-lectures will be surveys or discussions of open problems. Students should register to give mini-lectures upon an announcement from the TA. The registration is between the Thursday when the schedule is published and until Friday at midnight. Registration will be done using a shared spreadsheet. Generally, the first student to pick a certain mini-lecture will give it.

1. August 30, Workshop on approximation basics, including constraint satisfaction problems, label-cover, unique label-cover
2. September 13, Workshop on approximation algorithms based on linear programming
3. September 20, Workshop on approximation algorithms based on semidefinite programming
4. September 27, Workshop on hierarchies, connections to proof complexity and communication complexity
5. October 4, Workshop on PCP basics, including expanders and randomness efficient sequential repetition, transformation to two queries, degree reduction
6. October 18, Workshop on linearity testing and quadratic testing, Fourier, locally testable and decodable codes
7. October 25, Workshop on sum-check and its many applications, low degree testing,
8. November 1, Workshop on composition and code concatenation
9. November 8, Workshop on gap amplification
10. November 15, Workshop on parallel repetition
11. November 22, Workshop on long code and optimal inapproxmability

7 Final Grade

Every student should give *at least* 4 talks in 4 different workshops spread around the semester. Depending on enrollment, and to be determined within the first couple of weeks of the term, the number of required lectures may increase, up to 6 different talks during the semester. Each talk will be graded on correctness and on delivery on a scale of A=3/B=2/C=1/F=0. The final grade will be determined as follows:

- 70% Correctness of workshop talks
- 15% Delivery of workshop talks
- 15% Participation