CS 791 - Approximation Algorithms

K. Subramani
LDCSEE,
West Virginia University,
Morgantown, WV
ksmani@csee.wvu.edu

1 General Information

1. Meeting Times: Mon-Tu, 11:00 am – 12:15 pm
   Location: 209, MRB.
2. Contact Information: 749 ESB, k.subramani@mail.wvu.edu.
3. Office Hours: By appointment.
4. Prerequisites: Exposure to algorithm design, **NP-completeness** and Mathematical Programming.
5. Textbook - [Vaz02] is the official course textbook.
6. URL - `http://www.csee.wvu.edu/~ksmani/courses/fa14/approx/approx.html`.
7. Assessment:
   (a) Homework Assignments (3) - There will be three homework assignments, as per the following schedule:

Assignment Date	Submission Date
01/28	02/04
03/04	03/11
04/03	04/10

Table 1: Homework Schedule

   Each homework is worth 20% (for a total of 60%) of your grade.

   (b) Presentation (1) - You will be required to present one topic which will be decided through discussions with the instructor. The presentation is worth 20% of your grade.

   (c) Research - You are expected to engage in independent research on a problem of your choosing (or a critique of existing research), related to the topics of this course. This research is to be summarized in a report, to be handed in on the last day of class and is worth 20% of your grade. *If your research results in publishable work, then you will be assigned an* **A** *in the course, regardless of your performance on the other assignments.* Alternatively, you will be handed a fourth homework assignment, on April 24, which you will need to turn in by May 1.

   (d) Final - The final will be no final exam.

   (e) A maximum of 10 bonus points will be awarded for class performance

8. Grade Boundaries
   (a) **A**: 75 and up
   (b) **B**: $65 - 74$
   (c) **C**: $50 - 64$
   (d) **D**: $45 - 49$
   (e) **F**: $0 - 44$
9. Grading policy - If you have any questions about the grading, you must contact the intructor within two days of your paper being returned.
10. Makeup Policy - If for some reason, you are unable to attend a test or an exam, please meet me at the earliest and I will set an alternate date.
11. **Course Objectives** - The objectives of this course are to:
    (a) Re-introduce the notions of **NP-completeness** and **NP-hardness**.
    (b) Develop the notion of approximation algorithms.
    (c) Develop lower bounding techniques using combinatorial algorithms.
    (d) Develop linear programming based techniques for lower bounds.
    (e) Introduce lower bounding through semidefinite programming.
    (f) Develop the notion of inapproximability.
12. **Learning Outcomes** - Upon successful completion of this course, students will be able to:
    (a) Analyze an **NP-hard** problem from the perspective of approximability.
    (b) Establish non-trivial lower bounds for **NP-hard** minimization problems.
    (c) Attempt several techniques towards the design of approximation algorithms for a given problem.
    (d) Identify the possibility of inapproximability for a given problem.

2 Syllabus Sketch and Weekly Schedule

2.1 Introduction

Design and analysis of algorithms, measures of efficiency, size of input, time complexity, space, complexity, reducibility, measures of accuracy, additive and multiplicative errors, approximation algorithms.

2.2 Probability Theory Overview

Sample Space, Probability space, Random Variables, Expectation, Variance, Linearity of expectation and partial linearity of variance, Common distributions, Tail bounds.

2.3 Complexity Theory Overview

Problems as languages over $\Sigma^* = \{0, 1\}^*$, certificates and the class **NP**, computation tree of a non-deterministic computation, reductions (Turing and Karp) and **NP-completeness**, optimization problems and **NP-hardness**, approximation factor preserving reductions, self-reducibility, Randomized complexity classes **RP**, **coRP** and **ZPP**.

2.4 Linear and Integer Programming

Linear Programming forms, duality, Farkas' Lemma, relaxing a constraint, Integer Programming forms, binary integer programs, difficulty of proving existence of short certificates, modeling disjunction, implication, k out of m constraints, vertex cover, independent set, clique, relation between LP and IP, solving IPs through branch and bound, cutting planes and implicit enumeration.

2.5 Vertex Cover

Problem definition, unweighted (cardinality) and weighted versions, the maximum degree heuristic, matchings, the maximal matching heuristic for the unweighted case, tightness analysis.

2.6 Set Cover

Problem definition, relation to vertex cover, the maximum frequency heuristic, amortized analysis for the cardinality version, amortized analysis for the weighted version, tightness analysis. The layering approach for vertex cover.

2.7 Steiner Tree

Problem definition, The metric case, equivalence of the general problem to the metric case, analysis of a minimum-spanning tree heuristic, tightness analysis.

2.8 The Traveling Salesman Problem (TSP)

Problem definition, Variants, The Metric TSP problem, Analysis of the MST heuristic, Christofides' heuristic, tightness analyses.

2.9 Multiway Cut and k-Cut

Minimum cuts and maximum flows, definition of the Multiway Cut problem, Isolating cut, Approximation algorithm and analysis, tightness analysis. Definition of the k-Cut problem, Gomory-Hu trees, cuts corresponding to edges of trees, approximation algorithm and analysis, tightness analysis.

2.10 k-Center

Problem definition, A greedy 2-approximation algorithm, Parametric pruning, Dominating set of a graph, square of a graph, the graph-square based algorithm and its analysis, tightness analysis, inapproximability. The weighted k-center problem, weighted dominating set, approximation algorithm and analysis, tightness analysis.

2.11 Feedback Vertex Set

Problem definition, introductory linear algebra, Cyclomatic weighted graphs, Layering applied to the feedback vertex set problem, tightness analysis.

2.12 Shortest Superstring

Problem definition, a $2 \cdot H_n$ approximation algorithm through set cover, a factor 4 algorithm, a factor 3 algorithm, tightness analysis.

2.13 Binary Knapsack

Problem definition, the fractional knapsack problem, greedy algorithm for the fractional case, a dynamic programming algorithm for the discrete case, the fractional approach to the discrete case, a factor $\frac{1}{2}$ algorithm, a fully polynomial time approximation scheme, discussion of strong **NP-hardness** and the existence of fully polynomial time approximation schemes.

2.14 Bin Packing

Problem definition, offline and online versions, lower bounds on the online version, the Next Fit, First Fit and Best Fit heuristics, the First-Fit Decreasing heuristic, an inapproximability result, an asymptotic PTAS.

2.15 Minimum Makespan Scheduling

Problem definition, a factor 2 algorithm, The LPT rule and analysis, restricted bin-packing, reduction from makespan to restricted bin-packing, a PTAS for minimum makespan.

2.16 Introduction to LP-Duality

The Linear Programming duality theorem, Min-max relations and LP-duality, two fundamental techniques - LP rounding and primal-dual schema, comparison of the two techniques, notion of integrality gap.

2.17 Set Cover via Dual Fitting

Dual-fitting-based analysis for the greedy set cover algorithm.

2.18 Rounding Applied to Set Cover

A simple rounding algorithm, Randomized rounding, Half-integrality of vertex cover.

2.19 Set Cover via Primal-Dual Schema

Overview of the schema, Primaldual schema applied to set cover.

2.20 Maximum Satisfiability

A simple randomized algorithm, Dealing with large clauses, Dealing with small clauses via LP-rounding, a $\frac{3}{4}$ factor algorithm, Derandomizing via the method of conditional expectation.

2.21 Semidefinite Programming

Strict quadratic programs and vector programs, properties of positive semidefinite matrices, the semidefinite programming problem, a randomized rounding algorithm, improving the guarantee for MAX-2SAT.

I would like to reiterate that the above enumeration is a sketch of topics that I intend to cover. Due to various reasons, certain topics may be dropped or added. In such cases, advance notice will be given.

3 Topics for Presentation

The topics in the following list will be covered by student presentations.

1. Scheduling on unrelated Parallel Machines.
2. Multicut and Integer Multicommodity Flow in Trees
3. LP-based approach for Multiway Cut.
4. Steiner Forest and Steiner Network.
5. Facility Location.
6. Hardness of Approximation.

4 Inclusivity Statement

The West Virginia University community is committed to creating and fostering a positive learning and working environment based on open communication, mutual respect, and inclusion.

If you are a person with a disability and anticipate needing any type of accommodation in order to participate in this class, please advise me and make appropriate arrangements with the Accessibility Services (293-6700). For more information on West Virginia University's Diversity, Equity, and Inclusion initiatives, please see http://diversity.wvu.edu.

References

[Vaz02] Vijay Vazirani. *Approximation Algorithms*. Springer Science Publishers, 1^{st} edition, 2002.