

IEOR E8100 Topics in IEOR: Introduction to Discrete Optimization

Instructor: Yuri Faenza, yf2414@columbia.edu

Description. Discrete Optimization models a large number of questions arising in various area of engineering and mathematics: from problems on graphs, to production and scheduling problems, and much more. The most common way to cast such problems is as Integer Programs (IPs), where we aim at maximizing a linear objective function over the set of integer points that satisfy a finite number of linear inequalities. Since, in general, IPs are very hard to solve, much of the classical and modern research has been focused on algorithms to produce solutions of good quality to those problems. The goal of this course is to provide an introduction to IPs and to geometric approaches for their solution.

A very powerful technique to attack IPs is to formulate them – approximately or exactly – as linear or semidefinite optimization programs, and then solve the latter using classical algorithms. Most of the course will be spent in introducing a number of techniques to produce those formulations, and apply them to classical IPs like matching, flow problems, TSP, knapsack, etc.

We will moreover investigate Lenstra’s classical algorithm for solving integer programming. His algorithm relies on the theory of lattices, which will also be presented in this course.

Prerequisites. Linear algebra, linear programming.

Grading.

35% Assignments.

15% Scribing.

50% Final project.

Textbooks. Part of the course will be based on following books:

- A. Schrijver. Combinatorial Optimization. Springer, 2003.
- M. Conforti, G. Cornuéjols, and G. Zambelli. Integer Programming. Springer, 2014.

However, all the topics covered in the course will be covered by lecture notes or by scribes by the student. Each student taking the class for credits is expected to scribe one lecture.

Tentative schedule.

- Introduction to integer programming and the combinatorics of polyhedra.
- Tractable polytopes: total unimodularity, total dual integrality. Matching and flow polytopes.
- Hard polytopes: TSP polytope, methods to enhance formulations.
- Extended formulations.
- Hierarchies and approximate formulations.
- Lattices and Lenstra’s algorithm for Integer Programming.