CO 450/650: Combinatorial Optimization

Fall 2019: TTh 2:30-3:50pm, DWE 3516

Instructor: Chaitanya Swamy

Course overview

Combinatorial optimization is a field that combines techniques from combinatorics, linear programming, and the theory of algorithms to solve optimization problems over discrete structures. An instance of such a problem typically involves a given finite space of solutions, and a rational cost function that assigns a cost to each solution, and the goal is to find a minimum-cost solution efficiently (i.e., in time polynomial in the number of bits required to encode the input). Since the space of solutions can be quite large, simply enumerating over all solutions will not usually yield an efficient algorithm. So one needs to gain a suitable level of understanding about the problem structure, and leverage this understanding to design an efficient algorithm. For example, consider the problem of finding a minimum-weight set of edges of a graph whose removal disconnects two specified nodes s and t. Although the number of such edge-sets (called s-t cuts) can be exponential in the input size, one can exploit the problem structure to devise very efficient algorithms.

In this course, we will investigate a variety of well-behaved discrete-optimization problems, i.e., problems for which efficient exact algorithms exist. We will also see some applications of the techniques we develop in finding provably near-optimal solutions to problems that are hard to solve exactly. Topics covered include:

- Spanning trees
- Maximum flows and minimum cuts
- Matchings and generalizations
- Matroids and matroid-optimization problems
- Applications to approximation algorithms

Expectations and prerequisites

Apart from general mathematical maturity, students will be expected to have working familiarity with the theory of linear programming (what is a linear program, duality theory, geometry of linear programs), and basic graph theory terminology and notation (e.g., what are graphs, bipartite graphs, directed graphs, paths, cycles, trees, connectivity). Some pointers to material where you can refresh/gain knowledge of these topics are:

- **Linear programming**: Appendix A (pages 325–335) of *"Combinatorial Optimization"*, Cook, Cunningham, Pulleyblank, and Schrijver (does not discuss geometry of LPs). This will also be the main reference text for the course.

  Some notes on linear programming will also be posted on the LEARN page for the course.
- **Linear programming and graph theory**: Chapters 1–3 (pages 1–64) of *"Combinatorial Optimization"*, Korte and Vygen; available online at: https://link.springer.com/book/10.1007%2F3-540-29297-7.