MATH 7234 Optimization and Complexity **Spring 2021**

Northeastern University

Instructor: Dr. Oana Veliche
Contact: Lake Hall 543, x5537, o.veliche@northeastern.edu
Zoom Office hours: Mondays 7:00 – 8:00 pm and Wednesdays: 3:00 – 5:00 pm
Lectures: Tuesdays: 6:00 – 9:00 pm, in RY 159

Recommended Texts:

- *Combinatorial Optimization: Algorithms and Complexity*, by Papadimitriou and Steiglitz, Dover
- *Convex Optimization*, by Stephen Boyd and Lieven Vandenberghe

Course Description: This course deals with theory and methods of maximizing and minimizing solutions to various types of problems. We begin with examples of combinatorial problems of the following types: mixed integer programming problems ($\mathcal{MIP}$); pure integer programming problems ($\mathcal{I}P$); Boolean programming problems; linear programming problems $\mathcal{LP}$. Solving these problems will come later. We will briefly discuss the Simplex Algorithm for (LP). We will discuss the relationship between an ($\mathcal{LP}$) problem and its ($\mathcal{NLP}$) problem, and the Duality Theorem. In order the gain an overview, we will then go back to a very general class of functions, continuous functions, and quickly specialize to differential functions, and then to linear functions. We will also specialize from arbitrary subsets of $\mathbb{R}^n$ (n-space) as domains to convex subsets and then to polyhedral subsets. At the end of the process, we are back in the realm of Linear Programming ($\mathcal{LP}$). On the other hand, backing up to differentiable nonlinear functions we will look at Non-Linear Programming ($\mathcal{NLP}$). When the domains are convex sets we have convex programming. We will study the Kuhn-Tucker conditions for optimality for non-linear functions. We will use the Branch-and-Bound method for solving Integer Programs. In the last 3 to 5 weeks of the course we discuss complexity of algorithms. We focus on the problem classes $\mathcal{P}$ (problems with polynomial-time algorithms) and $\mathcal{NP}$ (problems with non-deterministic polynomial-time algorithms), and discuss Turing machines. We develop the notion of $\mathcal{NP}$-completeness, and establish that certain well-known problems are $\mathcal{NP}$-complete i.e. if they have polynomial-time algorithms then so do all the problems in $\mathcal{NP}$. It is for this last portion of the course that the (inexpensive) Dover text is recommended. If time permits, additional topics will be introduced.

Homework: It will be assigned and collected weekly. You may discuss the problems with your colleagues, but your work should be your own.

Tests: There will be 2 or 3 tests given this semester based on the material covered. You will be asked to reproduce proofs, give definitions and examples, and solve exercises.

Grading: Homework (70%), Tests (30%)