

Optimization and Complexity: Course Description

Prof. Ramras

Mathematics G234

Spring, 2009

Office: 547 Lake

Phone: x5651

e-mail: ramras@neu.edu

Office Hours: Tuesdays 4:30–5:30, and at other times by appointment

Time of class: Tuesdays, 5:50:00–9:00:00 (with a 15 minute break around 7:30pm)

Room: TBA

Required Text:

Introduction to Operations Research, 8th ed., by Hillier and Lieberman, McGraw Hill

Recommended Text:

Combinatorial Optimization: Algorithms and Complexity, by Papadimitriou and Steiglitz, Dover.

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 (MIP); pure integer programming problems (IP); Boolean programming problems; linear programming problems (LP). Solving these problems will come later.

We will discuss the Simplex Algorithm for LP (this is gone into in greater detail in the text "Into to Operations Research"). We'll discuss the relationship between an LP problem and its dual LP problem, and the Duality Theorem. In order to gain an overview, we will then go back to a very general class of function, continuous functions, and quickly specialize to differentiable functions, and then to linear functions. We also specialize from arbitrary subsets of \mathcal{R}^n (n -space) as domains to convex subsets and then to polyhedral subsets. At the end of this process, we are back in the realm of Linear Programming (LP). On the other hand, backing up to differentiable nonlinear functions we will look at Non-linear Programming (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 P (problems with polynomial-time algorithms) and NP (problems with non-deterministic polynomial-time algorithms), and discuss Turing machines. We develop the notion of NP completeness, and establish that certain well-known problems are NP -complete

i.e. if they have polynomial-time algorithms then so do all problems in NP . It is for this last portion of the course that the (inexpensive) Dover text is recommended.

There will be no exams. Your grade will be based on a number of problem sets.