Computational Optimization
ISE 407

Introduction

Dr. Ted Ralphs

Administrivia

- Class Meeting Time
- Introductions
- Office Hours
- TA

What is the motivation for this class?

- This class will introduce you to tools you will need to do computational research.
- We will touch on a wide range of topics that you should be familiar with before you start coding.
- We will not go be able to go into any of the topics we cover in much depth.
- It is expected that this introduction will be the beginning of your own investigation .
- As you do more coding and gain experience, you will quickly add to your repertoire.

What will this class be about?

- We'll talk about how computers work and how we can model them.
- We'll talk about how to use those models to analyze the efficiency of computational methods in theory.
- We'll discuss practical tools for implementing computational methods and talk about how they work.
- We'll discuss and analyze the implementations of data structures and algorithms.
- We'll discuss how to do the empirical testing and analysis that serves as practical alternative to theoretical analysis.
- The ultimate focus will be on developing the practical toolbox needed to implement and assess algorithms for solving optimization problems.

Topic Coverage

- Fundamentals of Computer Systems
- Models of Computation and Complexity
- Programming, Data Structures, and Empirical Analysis
- Algorithms
  - Combinatorial
  - Numerical

Prerequisites

- Strong undergraduate mathematics background
- Some programming experience
- Graph theory helpful
- Optimization background helpful

Unified Approach

- We will talk about the design and implementation process from end to end.
- As much as possible, we'll treat all subjects as a unified whole.
- We'll assume parallel platforms and algorithms from the outset.
- We'll try to emphasize modern hardware trends.
- Programming projects will build on each other.
- There is flexibility in the syllabus, let me know what you want to talk about!
- Technology is a moving target, if something seems out of date, let me know and we'll dive into it.
- I want this class to be relevant and useful.

What will I get out of this class?

- What you put into it . . . .
- In this course, we will raise a myriad of interesting computational issues.
- We won't have time to resolve them all.
- It is more important to have awareness of the issues.
- The best solutions may differ over time and with different platforms.

My approach to lectures

- There should be an active dialog between teacher and students
- This course will be loosely structured, so it's important that you ask questions!
- I will try to customize lectures them to the background of the students.
- Lectures will be an introduction to more in-depth reading.

More on lectures

- There will be a Web site for the class at

http://coral.ie.lehigh.edu/ ted/teaching/ie407

I will do my best to post the lecture slides there before the class so that you can prepare if you want.

All handouts for the class will be available at the site.

All readings will also be posted there.

Assignments and Exams (Tentative)

- There will be two types of assignments:
  - Programming
  - Written
- Assignments will be given approximately once every two weeks.
- There will be no formal written exams.
- There will be no written final.
- There will be a final project.

Submitting Assignments

- Assignments can be submitted through Google Classroom.
- For programming assignments, I would like well-commented, formatted source code.
- It would be helpful to have an accompanying write-up explaining the code.

Textbook

- There will be no single required textbook.
- There may be a few books that may be worth purchasing.
- There will be readings available on the Web site.
- Please feel free to seek out recent articles on the Web and let me know what you find.

Programming Environment

- Operating Systems
  - Unix
  - Windows
  - OS X
- Languages
  - C/C++
  - Matlab
  - Python

What Do You Want To See?

- GPU programming
- Containers (Docker)
- Continuous integration
- Version control (git and Github)
- Scripting and automation
- Running batch jobs
- Command line interfaces and bash scripting
- Makefiles and autotools
- CMake
- Jupyter Notebooks
- ...