Department of Computer Science/Industrial and Systems Engineering
Spring 2011

Tools and Environments for Optimization

(CS/ISyE 635)

Lecture: Monday and Wednesday, 2:30-3:45PM 1153 Mechanical Engineering Building

Jeff Linderoth
Office: 3226 Mech. Engr.
Office hours: Wednesday 4PM – 5:30PM and Friday 9:30-10AM (also by appt.)
Public calendar available at http://www.google.com/calendar/embed?src=jtlinderoth@gmail.com
Phone: 608-890-1931
E-mail: linderoth@wisc.edu
Web: http://homepages.cae.wisc.edu/~linderot/

Formulation and modeling of applications from computer sciences, operations research, business, science and engineering involving optimization and equilibrium models. Survey and appropriate usage of software tools for solving such problems, including modeling language use, subroutine libraries and web-based optimization tools and environments. Prerequisites: Comp Sci302, Math 340 or equiv. Knowledge of optimization modeling at the level of IE 323, or IE/CS 525 will be quite useful.

Course Materials

Required Texts

There are no required texts. A few texts and handouts will be made available on the course Learn@UW page. In particular, the entire GAMS User's Guide is available there and it will be a valuable reference for you.

Recommended Textx

The following texts are on reserve at the library:

H. P. Williams. *Model Building in Mathematical Programming*, 4th Edition (John Wiley & Sons, 1999). *This book is a bit outdated, but it is a* classic *reference on the importance of building good models for optimization problems*

Wayne L. Winston, Munirpallam Venkataramanan. *Introduction to Mathematical Programming : Operations Research.*, 4th Edition (Thomson, 2003).

Ron Rardin. *Optimization in operations research* (Prentice Hall, 1998).

In this course, we will primarily be using an algebraic modeling system called *GAMS*. It is a powerful commercial system for building optimization models, with hooks to state-of-the-art commercial solving engines.

Web Site

The course web site will be available through the the `Learn@UW` system: `https://learnuw.wisc.edu/`. This web site will be very important for the course, as we will use it for making assignments, turning in assignments, and posting lecture notes and other material. At first little material will be posted, but things will be added regularly during the course. Therefore please keep yourself aware of what is there.

Course Overview

This three-credit course introduces students to practical techniques for modeling and optimizing systems. Many companies and research programs are integrating optimization technology into their day to day operations. Manipulating models and optimization software and applying general operations research methodology in various contexts is becoming an increasingly sought after skill. This course is designed to teach students about "optimization in practice." The course involves learning about, using, and analyzing the results of state of the art optimization software. The student will learn how to design good models for realistic applications in engineering and the sciences. Each student will work to develop a "commercial strength" application of optimization technology.

Objectives

This course is intended to equip you with the following skills and insights:

1. The ability to write down an algebraic formulation of an optimization model that captures the main decision elements of practical problems.
2. The ability to categorize optimization models, and understand the implications of modeling on algorithm performance
3. To understand the tradeoff between model accuracy and tractability and to consider the feasibility of alternative design solutions
4. The ability to explain, at a non-technical level, how optimization may be applied to decision problems.
5. To become familiar with the operation of state-of-the-art optimization software, including parameters that may significantly affect software performance
6. Advanced knowledge of the GAMS Modeling System for building and solving practical optimization problems

There are other course objectives that will become clear as the course proceeds, but those listed above are the most important. The assignments given in this course are specifically designed to evaluate your progress in the above areas.

Requirements and Grading

Lectures

For instruction, the course will contain twice-weekly lectures. Learning is an interactive process. As such, attendance at lectures and discussion is mandatory, and a (small) portion of your grade will depend on your participation.

Homework

Students learn by doing. There will be (roughly) weekly homework assignments, and a large portion of the grade will depend on your performance on these assignments.

Exams

I am not currently planning on quizzes or final exams in the course.

Grading Scheme

The grade will be based on the following weighting of components.

- 5% Participation
- 65% Regular Homework Sets
- 30% Course Project

Course Statements

Course Project

The fundamental goal of the course is to equip you with the experience and background necessary to model and implement a real, large-scale optimization problem. As such, a significant portion of your grade will depend on a class project for building an optimization model or models. For the project, you will define, model, solve, analyze, and explain the solution(s) to a decision problem of your own choosing. Typically, students come to class having an idea of the project they would like to do.

In this semester, we are also offering special (monetary?) prizes to the top projects that contribute case studies to the NEOS website:
`http://www.neos-guide.org/NEOS/index.php/Case_Studies`.
One objective of the case studies for this class is to update the NEOS Guide case studies in an extendable way, thus also spreading the Gospel of Optimization to a broad audience. As examples of what you may be able to produce, the functional and useful case studies now are the Diet Problem, and the Sudoku Puzzle. For NEOS Case studies, the course project would consist of

- A web-wiki description of a problem, understandable to the general public.
- A model and submission that would solve the problem and provide some form of output/visualization. (The cooler the model and visualization, the more likely to win prizes).
- Possible updates to the submission to extend/learn about the capabilities of optimization

Such a case study would have to be incorporated onto the NEOS server. Expertise will be made available to you to help integrate this into NEOS.

Students may (self)-organize into teams of size at most three. More details about the project, including dates for deliverables, and possible ideas for projects, will be made available as the course progresses.

Homework and Case Study Submission Policy

Homework exercises will be assigned in most weeks as a means to help you understand the concepts and to give you practice in applying them. They will generally be due on Monday morning. Each assignment will be graded and the grades returned to you. Homework will be submitted electronically through the web site except when stated otherwise, which is another reason that it is essential to familiarize yourself with use of the web site at the beginning of the course.

In general, **late homework will receive no credit**. You typically will have one week or more to complete the homework assignment, so it is hard to imagine a reasonable excuse. Please complete your work in advance of the deadline, just in case "something comes up", and you are unable to complete the assignment at the last minute.

Learning is a collaborative process, so you are encouraged to verbally discuss the course work, including homework, with other students. However, work that you hand in *must have been prepared by you alone*. Further, you must cite all collaborators, as described in the section on "Use of External Sources" below.

Computing

You will need to make use of the GAMS, and information about obtaining and installing GAMS will be given during the course. We have obtained licenses for students to use GAMS on their personal Windows-based or Mac OSX computers. For students in CS, who have access to the Linux workstations, I can work to make GAMS available for you there as well.

I will work with all students to ensure that they have adequate access to computing and licenses for the software. If you have any troubles, please contact me.

Academic Integrity

You are encouraged to share ideas with each other on class assignments **orally**. However, you must ultimately demonstrate your understanding of the material by writing up your own solutions without the help of other students or their written work, including source code of the models you build. You may discuss the assignments with your classmates. However, you may not share any code, copy solution from another person, or carry out an assignment together. Discussion should only involve verbal communication. All assignments need to be written up entirely separately.

Submitting someone else's work as your own is academic misconduct. Such cheating and plagiarism will be dealt with in accordance with University procedures (see the Academic Misconduct Guide for Students: `http://www.wisc.edu/students/misconduct.htm`.)

Use of External Sources

Extensive materials related to the topics we will be covering in this class are available on the Internet. You may use supplementary material to enhance your understanding of the course material. If you use external references of any kind, even if they are not quoted verbatim, **YOU MUST CITE THEM**! This rule will be strictly enforced and violations will be dealt with harshly. Again, I encourage you to seek outside sources, but **you must acknowledge the source of any ideas that are not your own**.

AT NO TIME ARE YOU ALLOWED TO OBTAIN, LOOK AT, OR CUT AND PASTE FROM THE SOURCE CODE OR GAMS MODEL OF ANY OTHER INDIVIDUAL, WHETHER THEY ARE IN THE CLASS THIS SEMESTER OR NOT. THIS INCLUDES GETTING SOURCE CODE FROM THE INTERNET.

YOU ARE EXPRESSLY FORBIDDEN FROM SHARING ELECTRONIC FILES OF ANY TYPE BY ANY MEANS WITH ANYONE.

You are all grown-ups. **Do not cheat**. If you have any question or concern about what constitutes cheating or improper collaboration, *please* contact me. If I suspect that you are cheating, you will make me sad. Then you will make me mad. Do not do this. If you don't think that I will know if you are copying or inappropriately using other people's work to do your assignments, you are wrong.

(Tentative) Chronological Syllabus

I have built in significant flexibility in the schedule, depending on our progress, and the interests of the students in the course. More detailed assignment listings will be posted at the `Learn@UW` course web site when they are known. Other topics not listed here that could be covered depending on student interest include

- Other tools: AMPL, Mosel, Spreadsheet, Pyomo, Matlab
- Other environments: Callable libraries, web-based optimization tools

Part I: **Linear Models** *1/19-2/14*

Basic linear programming modeling. Introduction to GAMS, including advanced GAMS modeling and syntax features: Sets (Ordered and Dynamic), geometric intuition and properties of solutions of linear programs. Goal programming for multiobjective optimization.

Part II: **Network Models** *2/16-2/28*

Network Flow, Shortest Path, Assignment Problems. Handling sparse networks within GAMS.

Part III: **Discrete Models** *3/2-3/30*

Formulating discrete decision problems with integer variables. The impact of modeling on solution techniques. Fixed charges, Algorithmic Modeling, Set Covering, Specially Ordered Sets (Types I and II). Special discrete models such as Traveling salesperson and quadratic assignment.

Part IV: **Nonlinear Models** *4/4-4/13*

Nonlinear models, quadratic programming, practical portfolio optimization, least squares models, global optimization, conic optimization

Part V: **Uncertain Models** *4/18-4/27*

Stochastic Programming, Robust Optimization

Part VI: **Miscellaneous Topics** *5/2-5/9*

Mixed Integer Nonlinear Programming, Second Order Cone Programming, Complementarity Problems, Game Theory, Other topics as time and interest allow