CS/ECE/ISYE524: Introduction to Optimization – Linear Optimization Models

Jeff Linderoth

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

January 29, 2024

Today's Outline

- About this class.
- About me.
- About you.
- Modeling:
    - What is it?
    - Why should we care about this stuff?[1]
- A first model

[1] Besides, of course, the fact that you want to get a good grade

Class Overview

- As Snoop Dogg says, read the syllabus!
    - Posted on course Canvas page
    - It has many details about rules and regulations for the course
- Meeting Times: Monday-Wednesday 4:00PM-5:15PM
- Meeting Location: 1800 Engineering Hall

My Office Hours:

- Tuesdays: 9AM-10AM
- Fridays: 11AM-12PM
- By Appointment (last resort!)
    - My calendar is available at: `http://tinyurl.com/37upk9dp`
    - To make a meeting, please check my calendar, compare to your own free times, and suggest one or more times when we are both free

Course HomePage

- Canvas `https://canvas.wisc.edu/courses/397595`
- Lecture slides posted there. Posted as large modules, that may from time to time be updated.
- Announcements: You will be responsible for reading
- Organized into modules: One for each part of the course
- Homework assignments and submissions, grades, . . .
- Piazza Q&A Forum:
    - Available through Canvas. Check that you have access. If not, find our class signup link at: `https://piazza.com/wisc/spring2024/sp24isye524001`
    - I prefer that you *do not* email me (or the TAs) questions about the course. Rather, I prefer you ask them on the Q&A forum.
    - Unless the question is of a personal nature, I will copy email questions to the Q&A forum and answer them there.

Teaching Assistants

- Mr. Eric Brandt
  - **Office Hours:** 2:30-3:30PM MW, 3146 ME Bldg.
  - email: `elbrandt@wisc.edu`
- Mr. Sanjai Pushpa
  - **Office Hours:** 2PM-3PM RF, 3146 ME Bldg.
  - email: `pushpa@wisc.edu`

Brother, can you spare a dime?

- There are ≈ 190 of you in the class
- We have 40 hours of TA effort/week
- And (hopefully) some grading help
- The TAs and I are here to help you learn, but please be respectful of our time

Prerequisites

- **Comfortable with Programming:** `Comp Sci200/300` You don't necessarily need to know Java, but you do need to be comfortable with a computer programming language. If you are not, then this course is not really for you.
- **Comfortable with Linear Algebra:** `Math 340 or equiv`. You need to know a little bit of Calculus, be familiar with Matrix notation, and have the mathematical sophistication necessary to not be intimidated by symbols like $\sum$, $\forall$ and $\in$.
- **Computer Sophistication:** We will be learning/installing/using software Julia/JuMP in the course. You will be responsible for ensuring you have working coding environment. You also will (probably) need to learn LaTeX to type the mathematical description models you build.

Course Details

- Learning is better if you participate.
  - I will call on you during class. (Gasp!)
- Even though this is a huge class, I really would like the class to be interactive.
  - We are likely to spend time working through models
  - If you have a laptop, you may wish to bring it to class, in case we have interactive sessions.

Coursework Details

- Assignments
  - Expect to be relatively time-consuming
  - Largish assignments assigned every couple weeks or so. (Six assignments planned)
- Exams:
  - Two in-class midterm exams. (February 28, April 8)
  - Final exam (May 10)

Homework: Roughly Bi-weekly

- Homework submit through Canvas.
- Homework will be graded on a "spot" basis: Usually one or two problems randomly chosen and graded. The rest checked for complete effort (but not correctness).
- You are responsible for assessing your progress on the problem sets
- Problem solutions will be posted soon after due.
- The lowest homework score will be dropped
  - Purpose: Accommodate *all* reasons for missed work
- Late homework will be penalized 10%: And cannot be accepted once the solutions are posted.

Grading

- 30% Regular Homework Sets
- 20% Midterm #1
- 20% Midterm #2
- 30% Final Exam

- Syllabus has grade cutoff percentages, but I typically grade on a curve. Lowering grade cutoffs as necessary.
- The median performer in the class should get an `AB` grade, but below the median will likely get a `B` or lower.

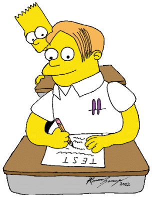
Course Texts

- None Required.

Recommended

- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. The book is available for free here: http://stanford.edu/~boyd/cvxbook/.
- H.P. Williams. *Model Building in Mathematical Programming*, 5th Edition. Wiley, 2013.
- R.L. Rardin. *Optimization in Operations Research*. Prentice Hall, 1998.

Cheater!!!!



Please don't cheat

- It will make me sad.
- And mad.

- Homework assignments can (and should!) be discussed together
- However, you must write up your answers independently!
- If too much "collaboration" on homework is noticed, I reserve the right to give all parties involved 0% on the assignment
- Using a LLM to do your homework is not an ethical use of that resource

Mathematical Topics

1. Linear and Network Programming ($\approx$ 9-10 lectures)
2. Convex Programming ($\approx$ 6 lectures)
3. Integer Programming ($\approx$ 8 lectures)
4. Stochastic Programming ($\approx$ 2 lectures)

Course Objectives

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. Use the Julia language and JuMP modeling package;
7. Have Fun (and work hard!)

Great Expectations

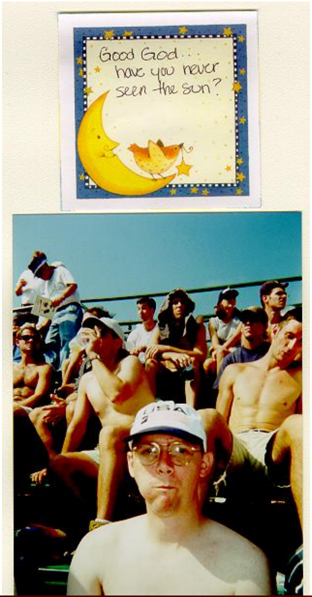
I am expected to...

- Teach
- Answer your questions
- Be at my office hours
- Give you feedback on how you are doing in a timely fashion

You are expected to...

- Learn
- Attend lectures and participate
- Do the problem sets
- Not be rude, if possible.
    - Sleeping, Talking, Showing up late
    - Cell Phones, Texting, Tweeting, Surfing
    - Leaving in the middle of lecture

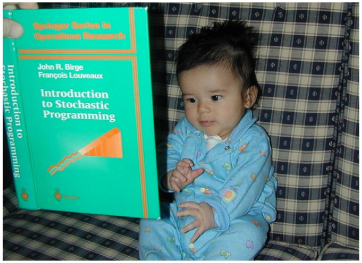
About me...



- B.S. (G.E.), UIUC, 1992.
- M.S., OR, GA Tech, 1994.
- Ph.D., GA Tech, 1998
- 1998-2000 : MCS, ANL
- 2000-2002 : Axioma, Inc.
- 2002-2007 : Lehigh University
- Research Area: Large-Scale Optimization
- (Current) Applications: Energy, Defense
- Married. One child, Jacob who is a junior in college. (Studying math)
- Hobbies: Golf, Human Pyramids, Integer Programming

Picture Time/About You

- I typically learn the names of all the students in my class.
- This may be too difficult for me this semester—I apologize!
- But I still would like to know about you—Please fill out the course interest survey on Canvas


Springer Series in Operations Research
John R. Birge
François Louveaux
Introduction to Stochastic Programming
Introduction to Stochastic Programming

Evil?!

- This is not an easy course
- Eric, Sanjai, and I really want you to learn the material and do well
- We will all have to work hard to make this happen
- With 190 students in the class, we cannot babysit you and provide a "hands-on" level of service. Please be professional.
- Please be patient with me—I have never taught a class this large before.

Decision Problems

- In this course you will learn to model decision problems

Decision Problems

1. How many clerks do I need at my grocery store?
2. How much capacity should I add to my telecommunications network?
3. How should I design my new airplane wing?
4. Is my new airbag design safer than the previous design?
5. What types of aircraft should fly each flight in a schedule?
6. What stocks should I buy?

- Warning! Not all of these questions are best answered by *optimization* models

Models

"A model should be as simple as possible and yet no simpler"
–Albert Einstein

Why Model

1. To get an answer!
2. From building a model, we can gain insight.
3. We can "experiment" with a model.

Types of models

- Physical
  - Airline wing design
- Abstract
  - Statistical: Time Series, Regression, etc...
  - Simulation
  - Economic
  - **Optimization!**

Common Fallacies

1. How can anyone possibly get and be confident in the data that makes up the model?
   - We're not. The analyst must understand the reality of the process to deduce whether the model solution makes sense
2. It's "too abstract"
   - It's not. The analyst must be able to explain *why* the solution approach is proper
3. If we got an answer on the computer, it must be right!
   - It's not. All models are wrong. But some are useful. (George Box)

The upshot!

(Optimization) models can be an important tool in a decision-making process.

Components of an Optimization Model

1. Decision variables
   - Variables representing the unknown quantities
2. Constraints
   - Requirements that all solutions must satisfy, (expressed algebraically)
3. Objective
   - A quantity that you would like to make as small or as large as possible.

Variables:

- x: Pounds of barley to purchase
- y: Pounds of hops to purchase
- z: Gallons of beer made

Constraints:

- $y \leq 2$
- $x \leq 8y$
- $z = 0.4x + 0.9y$

Objective:

- $\max z$

Another View at Model Components

1. Inputs
   - Sets. Used typically for algebraic models.
     - e.g., P: Set of products, I: Set of locations
   - "Numbers". These are called parameters. The parameters may be indexed over sets.
     - e.g., u_p: The maximum amount of product p available
2. Decision Variables
   - "Numbers you are allowed to change". It is the goal of the optimization to find the "best" values of these controls (or decision variables). Decision variables can also be indexed over sets
     - e.g., z_i: Gallons of beer to ship to location i
3. Outputs
   - These may be optimal values of the decision variables, or a derived value, such as the objective function value
   - e.g.: $\sum_{i \in \text{Madison}} z_i$

Categories of Optimization Models

- Linear vs. Nonlinear?
  - Are the functional relationships between decision variables linear functions or nonlinear functions?
- Convex vs. Nonconvex?
  - Are the functional relationships convex?
- Discrete vs. Continuous?
  - Must the decision variables take only discrete values?
- Deterministic vs. Stochastic?
  - Is uncertainty in the model explicity considered?

The upshot

- These categorizations have a significant impact on the tractability of an instance
- You should be able to categorize problem instances

Solving

- Actually involves gathering and processing data: Turning your model into an instance.
    - Model : A structure containing (algebraic) relationships between entities.
    - Instance : A combination of data and model that can be solved. (i.e – it has "numbers")
    - Spreadsheet models "blur" this distinction, that's why I don't like them very much.
- (Algebraic) Modeling languages are better!
    - Have hooks to solvers
    - Many have hooks to spreadsheets and databases
- We will use JuMP, a package in the Julia programming language to build our models.
- We will (try to) teach some best modeling practices about abstraction—separating model from data

524—Gateway to other optimization courses (usually more advanced)

- Linear programming (CS 525)
- Nonlinear optimization (CS 726, 730)
- Convex analysis (CS 727)
- Integer Optimization (CS 728)
- Stochastic/Dynamic programming (CS 719, 723)

Selected applied topics:

- Machine learning (CS 760, 761, 762)
- Optimal control (ECE 719, 819, 821)
- Robot motion planning (ME 739, 780)

(Many of these are cross-listed across other departments.)

Top Brass example

Top Brass Trophy Company makes large championship trophies for youth athletic leagues. At the moment, they are planning production for fall sports: football and soccer.

Each football trophy has a wood base, an engraved plaque, a large brass football on top, and returns $12 in profit.

Soccer trophies are similar except that a brass soccer ball is on top, and the unit profit is only $9.

Since the football has an asymmetric shape, its base requires 4 board feet of wood; the soccer base requires only 2 board feet. There are 1000 brass footballs in stock, 1500 soccer balls, 1750 plaques, and 4800 board feet of wood.

What trophies should be produced from these supplies to maximize total profit assuming that all that are made can be sold?

football soccer both

Top Brass data

Recipe for building each trophy

	wood	plaques	footballs	soccer balls	profit
football	4 ft	1	1	0	$12
soccer	2 ft	1	0	1	$9

Quantity of each ingredient in stock

	wood	plaques	footballs	soccer balls
in stock	4800 ft	1750	1000	1500

Top Brass Model Components

1. **Decision variables**
   - f: number of football trophies built
   - s: number of soccer trophies built
2. **Constraints**
   - $4f + 2s \leq 4800$ (wood budget)
   - $f + s \leq 1750$ (plaque budget)
   - $0 \leq f \leq 1000$ (football budget)
   - $0 \leq s \leq 1500$ (soccer ball budget)
3. **Objective**
   - Maximize $12f + 9s$ (profit)

Top Brass model (optimization form)

$$
\begin{aligned}
\underset{f,s}{\text{maximize}} \quad & 12f + 9s \\
\text{subject to:} \quad & 4f + 2s \leq 4800 \\
& f + s \leq 1750 \\
& 0 \leq f \leq 1000 \\
& 0 \leq s \leq 1500
\end{aligned}
$$

- This is an instance of a *linear program* (LP), which is a common type of optimization model.
- We have decision variables and parameters.

Top Brass model (generic)

$$\begin{aligned} \max_{f,s} \quad & c_1 f + c_2 s \\ \text{subject to:} \quad & a_{11} f + a_{12} s \leq b_1 \\ & a_{21} f + a_{22} s \leq b_2 \\ & \ell_1 \leq f \leq u_1 \\ & \ell_2 \leq s \leq u_2 \end{aligned}$$

- By changing the parameters, we create different *problem instance*.
- It's good practice to separate parameters (data) from the algebraic structure (model).
  - This makes it easy to try out different parameter settings, so see how they affect the optimal values of the decision variables.

Top Brass code (IJulia notebook)

```
using JuMP
m = Model()
@variable(m, 0 <= f <= 1000)      # football trophies
@variable(m, 0 <= s <= 1500)      # soccer trophies
@constraint(m, 4f + 2s <= 4800)   # total board feet of wood
@constraint(m, f + s <= 1750)     # total number of plaques
@objective(m, Max, 12f + 9s)      # maximize profit
```

Note: we did *not* separate the data from the model in this example! Next class, we will see how we can generalize the code.

Getting Started with Julia

1. Get up and running with Julia + IJulia + JuMP and jupyter notebooks, following the instructions posted on Canvas.
2. Work through tutorials for Julia and JuMP.
3. Load the file Top Brass.ipynb in IJulia and confirm that you can reproduce the results shown in class.
4. Try to obtain an academic license and install Gurobi, and link it to Julia, using the instructions in Top Brass.ipynb.
    1. Gurobi is a high-quality commercial code for linear and integer programming.
    2. You won't need Gurobi in class, but if you are solving large problems in the project, you may find it useful.
5. Experiment! Try changing the parameters and seeing if the solution still makes sense.

Julia Tutorials

- Noteworthy differences between Julia and other languages: `https://docs.julialang.org/en/v1/manual/noteworthy-differences/`
- Useful tutorial: `https://learnxinyminutes.com/docs/julia/`
- Official Julia documentation: `https://docs.julialang.org/en/v1/`
- JuMP: `https://jump.dev/JuMP.jl/stable/`
- More resources on course Canvas Page

Assignment #0

- Get and install Julia and JuMP in your working environment.
- Please post any installation issues on Piazza, so your students colleagues (and the TAs and I) can try to help.
- We have posted some introductory Julia notebook tutorials. You should go through these and following the instructions for turning in JuliaTutorialExercises.ipynb
- You should also ensure that you can correctly print out the notebook as PDF
  - jupyter nbconvert –to pdf notebook.ipynb
  - You may need to install pandoc and/or xetex
- First homework coming soon, so please try to get things working.