1. Introduction

- Logistics
- Optimization examples
- About the course
- Detailed example with code

Laurent Lessard (www.laurentlessard.com)

Logistics

- **Instructor:** Laurent Lessard (l.lessard@northeastern.edu)
  Office hours: by appointment
- Canvas for all course materials and discussion.
- Gradescope for turning in assignments (access via Canvas)

Prerequisites

- Exposure to a bit of linear algebra and calculus (Math 2341 or equivalent). We will review all relevant concepts as we need them.
- Some experience with numerical computing and writing short scripts (e.g. Matlab, Python)

For this course, we will use **Julia**. Julia is a programming language designed for scientific computing similar to Matlab and Python. It's open-source and super fast!

Coursework

- Homework (200): weekly assignments, a mix of theory questions and practical problem-solving (with Julia coding). Roughly 10 total assignments.
- Midterm 1 (250): TBA.
  Similar to Homework, but with no coding.
- Midterm 2 (250): TBA.
  Similar to Homework, but with no coding.
- Final project (300): You will work in groups of 3.
  More details to come...

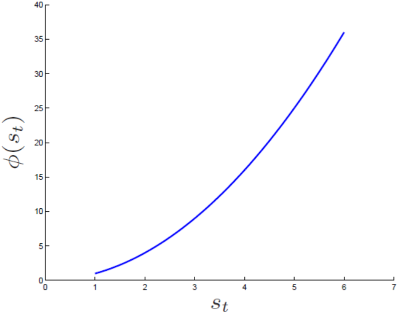
Textbook

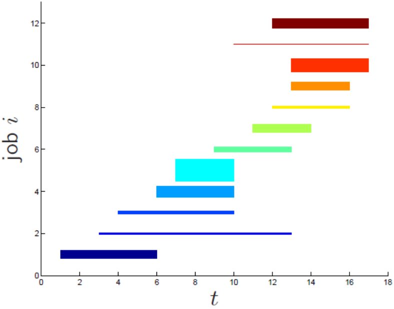
There is no required texbook for the course. However, there are plenty of good references. Here are a few:

- 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.

Ex. 1: Processor scheduling

- processor adjusts its speed s_t over time
- jobs must be scheduled. Each job has: arrival time, deadline, total work required.
- energy consumed $\phi(s_t)$ depends on speed

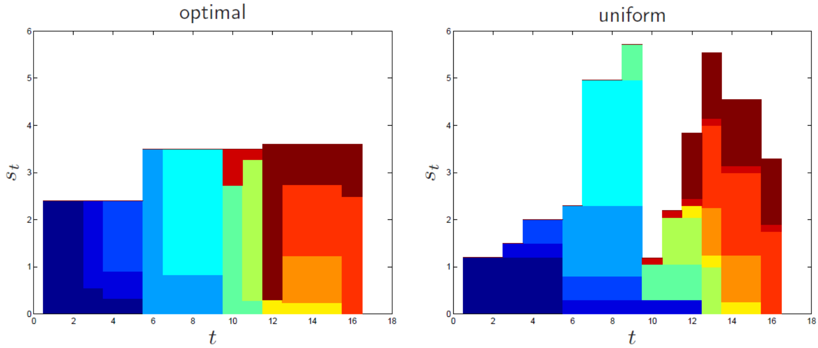




Source: S. Boyd (Stanford)

Ex. 1: Processor scheduling

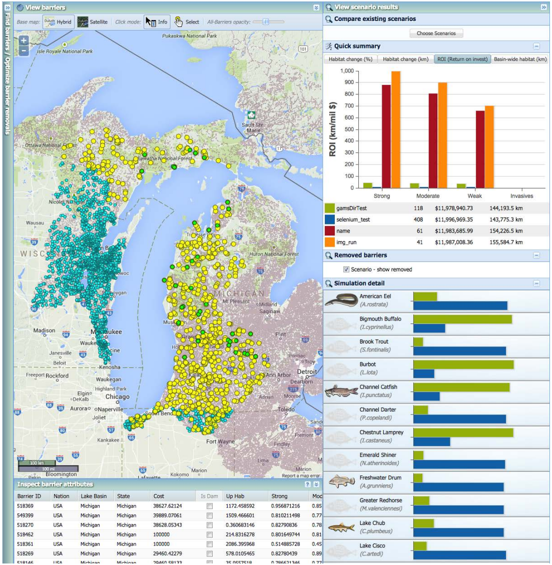
- uniform schedule: requires 204.3 energy.
- optimal schedule: requires 167.1 energy.
- similar to other types of scheduling problems!
  e.g. employees, power networks,...



Source: S. Boyd (Stanford)

Ex. 2: Fish and Wildlife Service

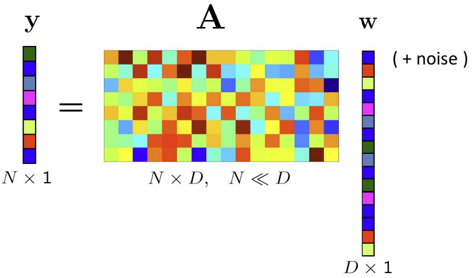
- Great lakes basin data visualization
- 250k interdependent barriers on the network of rivers
- Complex optimization for budget constraints, specific fish guilds, invasive species.



Source: M. Ferris (UW–Madison), Fishwerks.

Ex. 3: Compressive sensing

- N outputs (observations) caused by D inputs (features)
  - Disease prediction: $\mathbf{y}$ are patients, $\mathbf{A}$ are gene markers, or
  - Imaging: $\mathbf{y}$ are image benchmarks, $\mathbf{A}$ are MRI modalities.

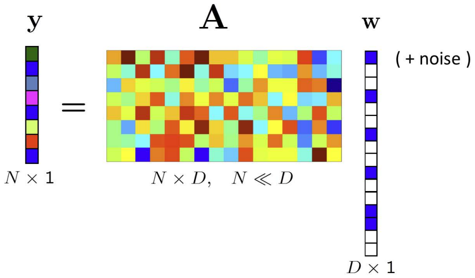


System is under-determined!

Source: R. Baranyuk (Rice) et.al.

Ex. 3: Compressive sensing

- N outputs (observations) caused by D inputs (features)
  - Disease prediction: $\mathbf{y}$ are patients, $\mathbf{A}$ are gene markers, or
  - Imaging: $\mathbf{y}$ are image benchmarks, $\mathbf{A}$ are MRI modalities.

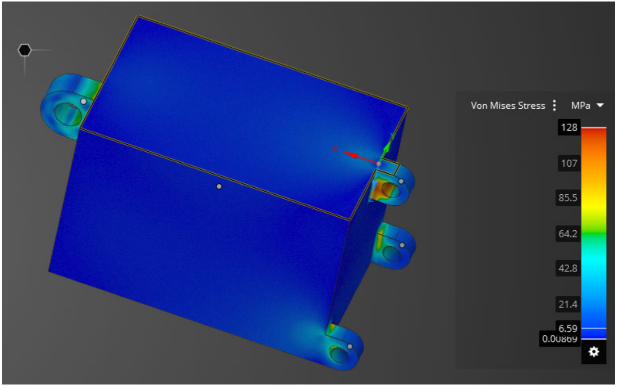


System is over-determined! (Look for a sparse solution)

Source: R. Baranyuk (Rice) et.al.

Ex. 4: Structural optimization

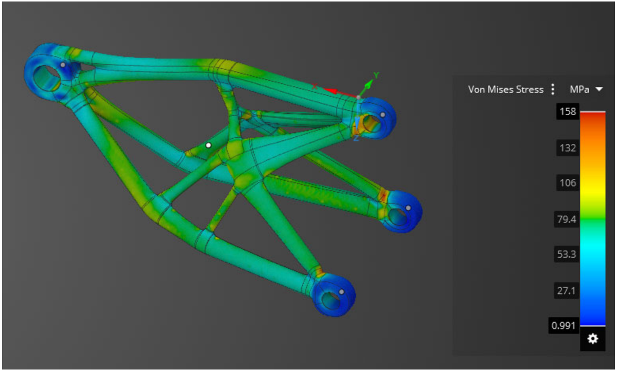
- Specify desired loading at given points
- Specify stress and strain limits of materials
- Goal: minimize total mass



Source: tensor.ro blog (link)

Ex. 4: Structural optimization

- Specify desired loading at given points
- Specify stress and strain limits of materials
- Goal: minimize total mass



Source: tensor.ro blog (link)

Course objectives (modeling!)

1. Write down an algebraic formulation of a optimization problem that captures key elements of the problem. We call this an **optimization model**.
2. Perform standard mathematical manipulations to enable you to categorize optimization problems.
3. Develop an intuition for different types of models, their solutions, and the implications and trade-offs involved.
4. Write code in Julia to solve these problems.
5. Learn to interpret solutions, validate, perform sensitivity analysis, explore trade-offs, etc.

Course topics (by model type)

1. Linear programs (LP)
2. Least squares and quadratic programs (QP)
3. Second-order cone programs (SOCP)
4. Convex programs (CP)
5. Integer programs (IP) and mixed-integer programs (MIP)
6. Nonlinear programs (NLP and MINLP)

This course is NOT

- An algorithms class. (but we will discuss and use them!)
- A coding class. (but we will write code!)
- A machine learning class. (no neural networks)
- A numerical methods class. (e.g., FEA, CFD)

Gateway to more advanced courses

- Linear programming (IE 4515, OR 6205, OR 7245)
- Convex analysis (OR 7270)
- Integer Optimization (OR 7240, EECE 5350)
- Nonlinear optimization (OR 7240, EECE 7323)
- Discrete algorithms (CS 3000, CS 4810)
- Complexity theory (CS 5800, CS 7800, MATH 7234)
- Computational methods (EECE 7345, EECE 7223)

Selected applied topics:

- Machine learning (CS 6140, CS 7140)
- Optimal control (EECE 7214)
- Logistics (OR 7310)
- Robot motion planning (EECE 5550)

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. At the moment 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)

Generic optimization model

min/maximize (variables)	(objective function)
subject to:	(constraint 1)
	(constraint 2)
	…

Generic optimization model (shorter)

$$\begin{aligned} \underset{\text{(vars/constraints)}}{\text{min/max}} \quad & \text{(objective function)} \\ \text{s.t.} \quad & \text{(constraint 1)} \\ & \text{(constraint 2)} \\ & \dots \end{aligned}$$

Top Brass optimization model

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

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

Top Brass optimization model (generic)

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

- By changing the parameters, we create different *instance*.
- It's good practice to separate parameters (data) from the algebraic structure (model).

Top Brass code (IJulia notebook)

Top Brass.ipynb

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