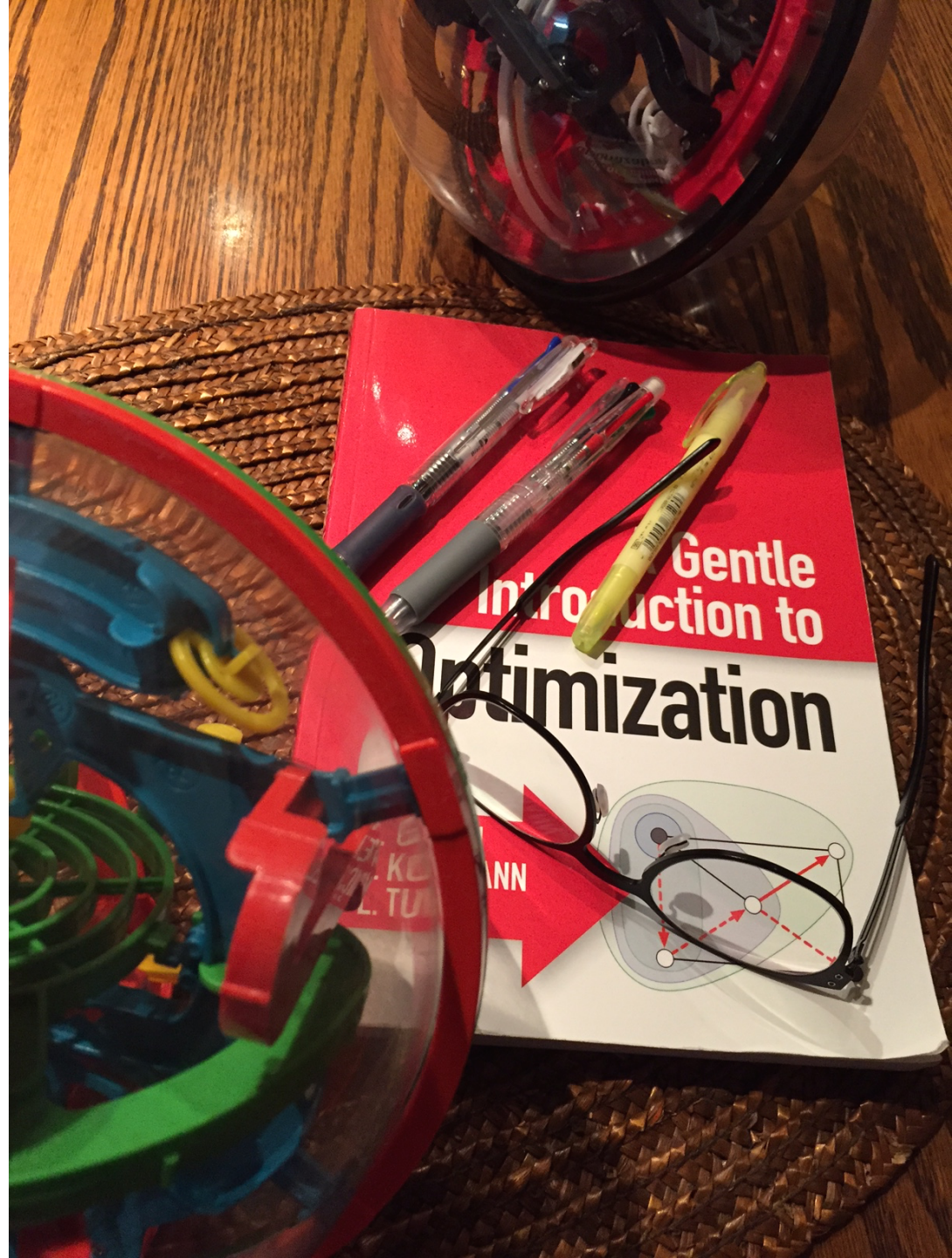
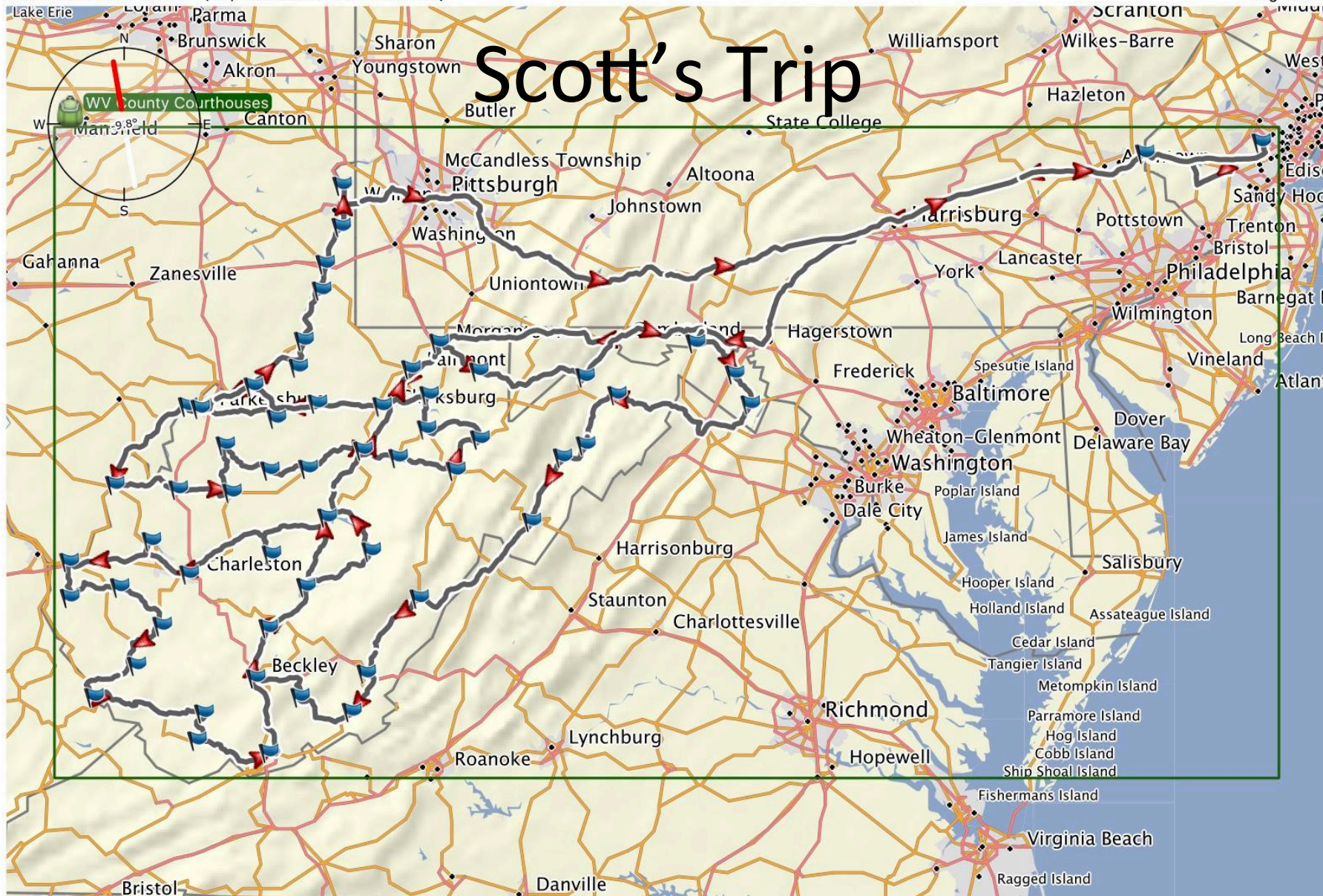


Optimization Theory

Assoc. Prof. Rodney
Van Meter
Lecture 1





Scott's Trip

City Navigator North America NT 2017.2

© Garmin Ltd. or its subsidiaries 2016.

0 50 100 150 200 250 mi

N36° 33.578' W73° 58.540'

WV County Courthouses

Factory Optimization

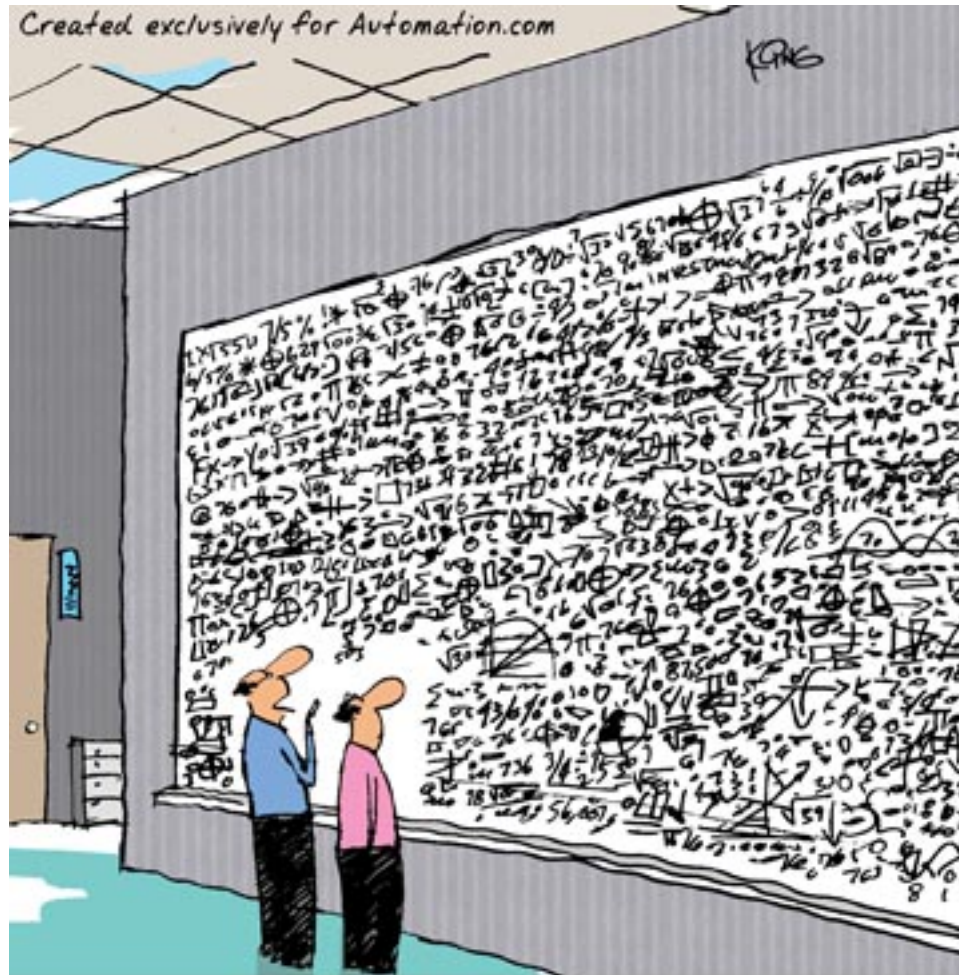


http://labs.blogs.com/its_alive_in_the_lab/2010/12/factory-layout-optimization-for-autocad-now-available.html



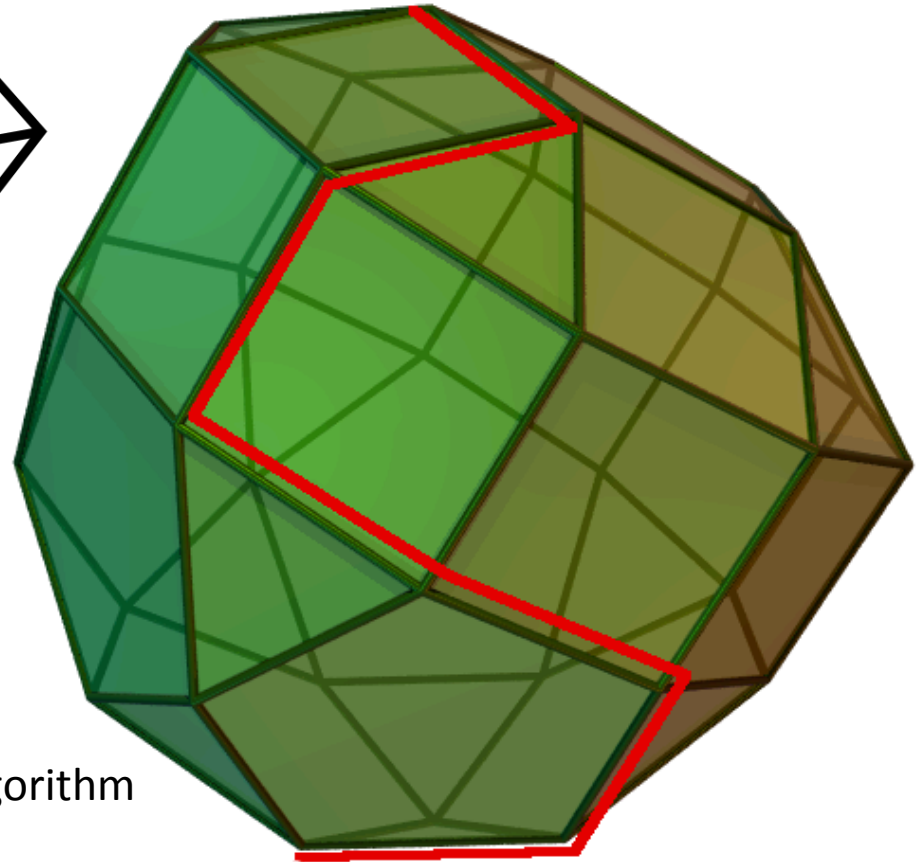
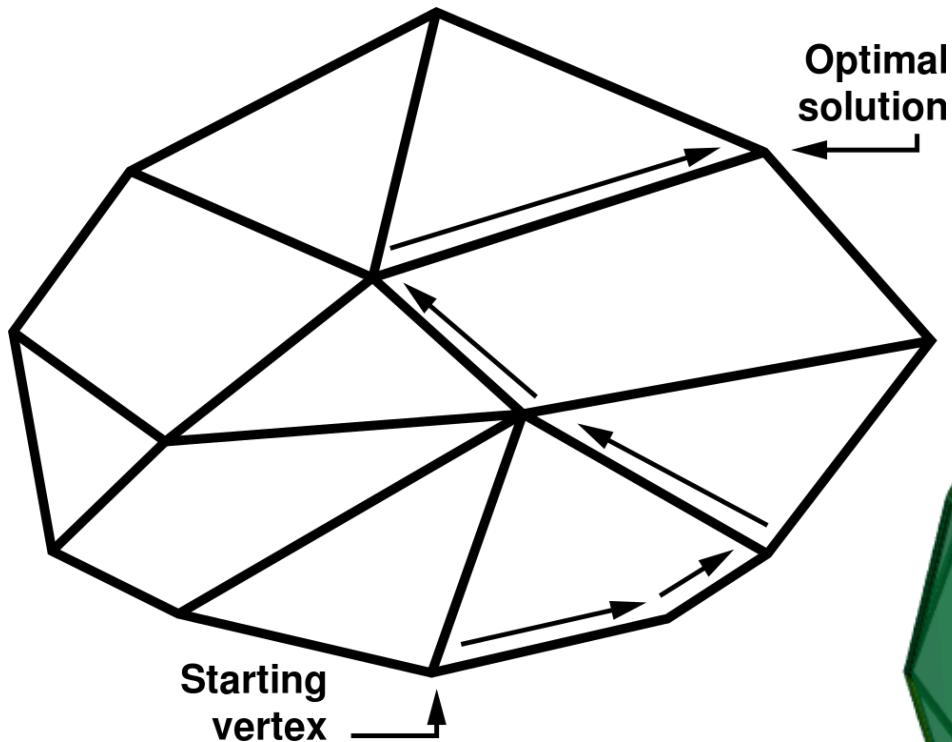
<http://www.eesc.st.keio.ac.jp/mech/contents01.html>

Factory Optimization



*“...and that, in simple terms, is my idea on
how to increase factory optimization.
any questions?”*

Simplex Algorithm

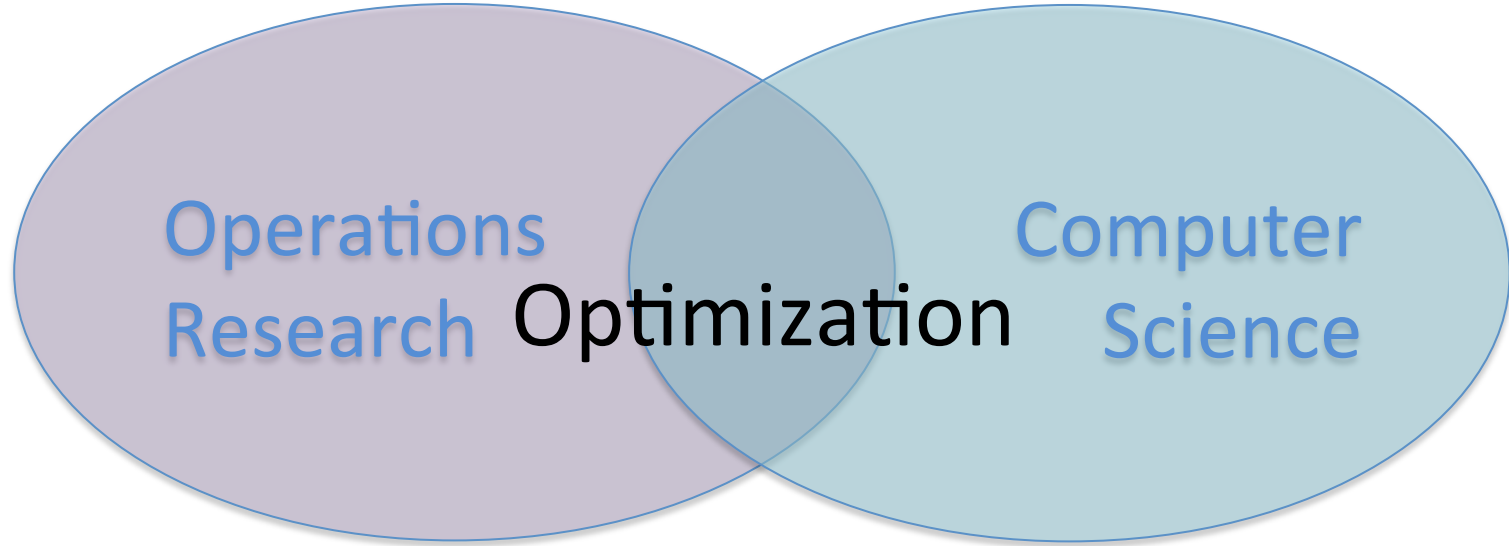


Images from
https://en.wikipedia.org/wiki/Simplex_algorithm

Optimization is Used for...

- Business
- Economics
- Manufacturing
- Networking (Internet)
- Robots (e.g., path planning)
- Biology
- ...pretty much every human activity.

As a Field, It's Related To...



Business
Economics
Manufacturing
Government Policy

Algorithms
Theory
AI
Robotics
Networking

...pretty much every human activity.

What's Hard About Optimization?

- Taking a rough, written description of a problem and turning it into a rigorous set of constraints.
 - Proving that the constraints match the problem!
- Solving the problem itself.
 - Computers are good for this, but only up to a point; where is that point?

Background

Mathematics:

- Vectors & Matrices: addition, multiplication, dot product, basis transformation
- Summation
- Set theory (basics)
- A little geometry
- Functions

Computer science:

- Basic concept of an algorithm: functions, variables, loops, conditions

Types of Programs (Problems)

- Linear Program (LP)
- Integer Program (IP)
- Nonlinear Program (NLP)

Linear Program

function: $f : \mathbb{R}^n \rightarrow \mathbb{R}$

affine function: $f(\vec{x}) = \vec{a}^\top \vec{x} + \beta$

linear function: $f(\vec{x}) = \vec{a}^\top \vec{x}$

linear constraint

(all vars on left, all constants on right):

$f(\vec{x}) \leq \beta$ or $f(\vec{x}) \geq \beta$ or $f(\vec{x}) = \beta$

“A *linear program* (LP) is the problem of maximizing or minimizing an affine function subject to a finite number of linear constraints.” GKT, p. 6

What Does All That Mean?

- (Let's do a little linear algebra review...)

Integer Programs

Same thing as LP, except that a non-empty subset of the variables must take integer values

- e.g., if you're packing a shipping container, can't put half a car in one container and half in another!
- In a *pure integer program*, they all have to be integers
- In a *mixed integer program*, only some of them do

Doesn't sound like a big deal, but it is; makes solving the problem *much* harder.

This Semester

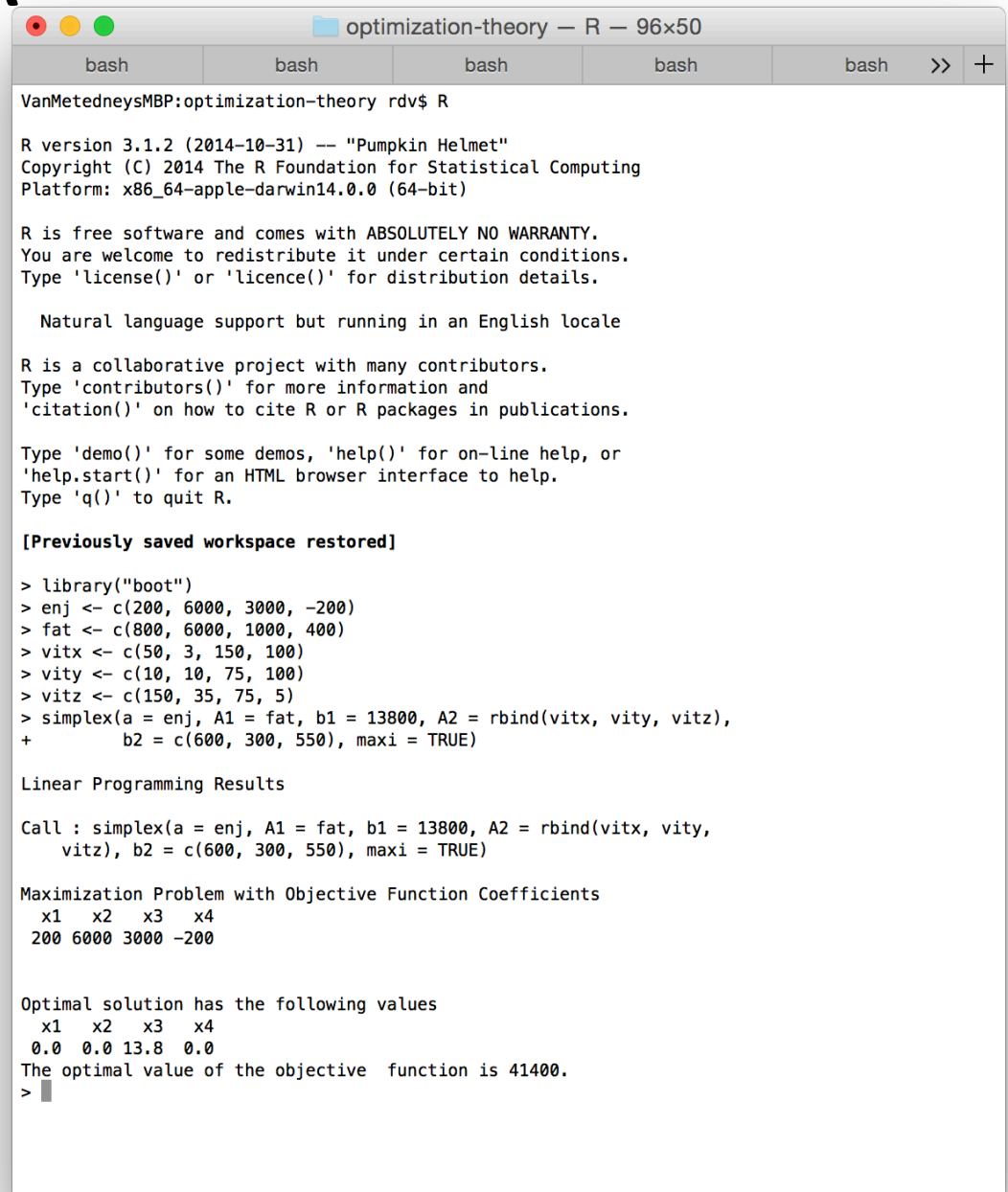
Week/ Chapter	Topic	Week/ Chapter	Topic
1/1	Introduction, Linear/Integer/ Nonlinear Problems, Business/ Manufacturing Examples	8/4.3	Duality Theory 2
2/1.4-1.5	Problems on Graphs and in Networking	9/6.1-6.2	Integer Problems 1
3/2.1-2.4	Solving Linear Problems: Certificates	10/6.3-6.4	Integer Problems 2
4/2.5-2.8	Solving Linear Problems: the Simplex Algorithm	11/App. A	Computational Complexity
5/3.1	Duality Examples 1: Shortest Path	12/7	Nonlinear Optimization
6/3.2	Duality Examples 2: Minimum Cost Perfect Matching	13/none	Problems in AI and Machine Learning
7/4.1-4.2	Duality Theory 1	14/none	Final Presentations

n.b.: I will be gone on Oct. 18. Class will be rescheduled to Nov. 11.

Programming: R

We will be using R for some problems in this class.

This is a “data science class”, not a “programming class”, but you will be using the computer as a *tool*. Some problems will be too big to do by hand.



```
optimization-theory — R — 96x50
bash bash bash bash bash >> +
VanMetedneysMBP:optimization-theory rdrv$ R

R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin14.0.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> library("boot")
> enj <- c(200, 6000, 3000, -200)
> fat <- c(800, 6000, 1000, 400)
> vitx <- c(50, 3, 150, 100)
> vity <- c(10, 10, 75, 100)
> vitz <- c(150, 35, 75, 5)
> simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity, vitz),
+         b2 = c(600, 300, 550), maxi = TRUE)

Linear Programming Results

Call : simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity,
vitz), b2 = c(600, 300, 550), maxi = TRUE)

Maximization Problem with Objective Function Coefficients
  x1  x2  x3  x4
200 6000 3000 -200

Optimal solution has the following values
  x1  x2  x3  x4
0.0 0.0 13.8 0.0
The optimal value of the objective function is 41400.
> █
```

Grading

- 40% homeworks
- 40% final project (presentation TBD)
- 20% class participation (means I need to know your name!)

Homeworks

- Target is 1-2 hours/week of regular homework almost every week
- Most will be exercises from the textbook

Final Project

Examples:

- Describe & optimize a significant real-world problem (definition of “significant” TBD)
- Implement simplex algorithm in language of your choice
- Tackle some problems from chapters we didn't cover (esp. max cut min flow)

Est. 10-20 hours (about half of total homework)