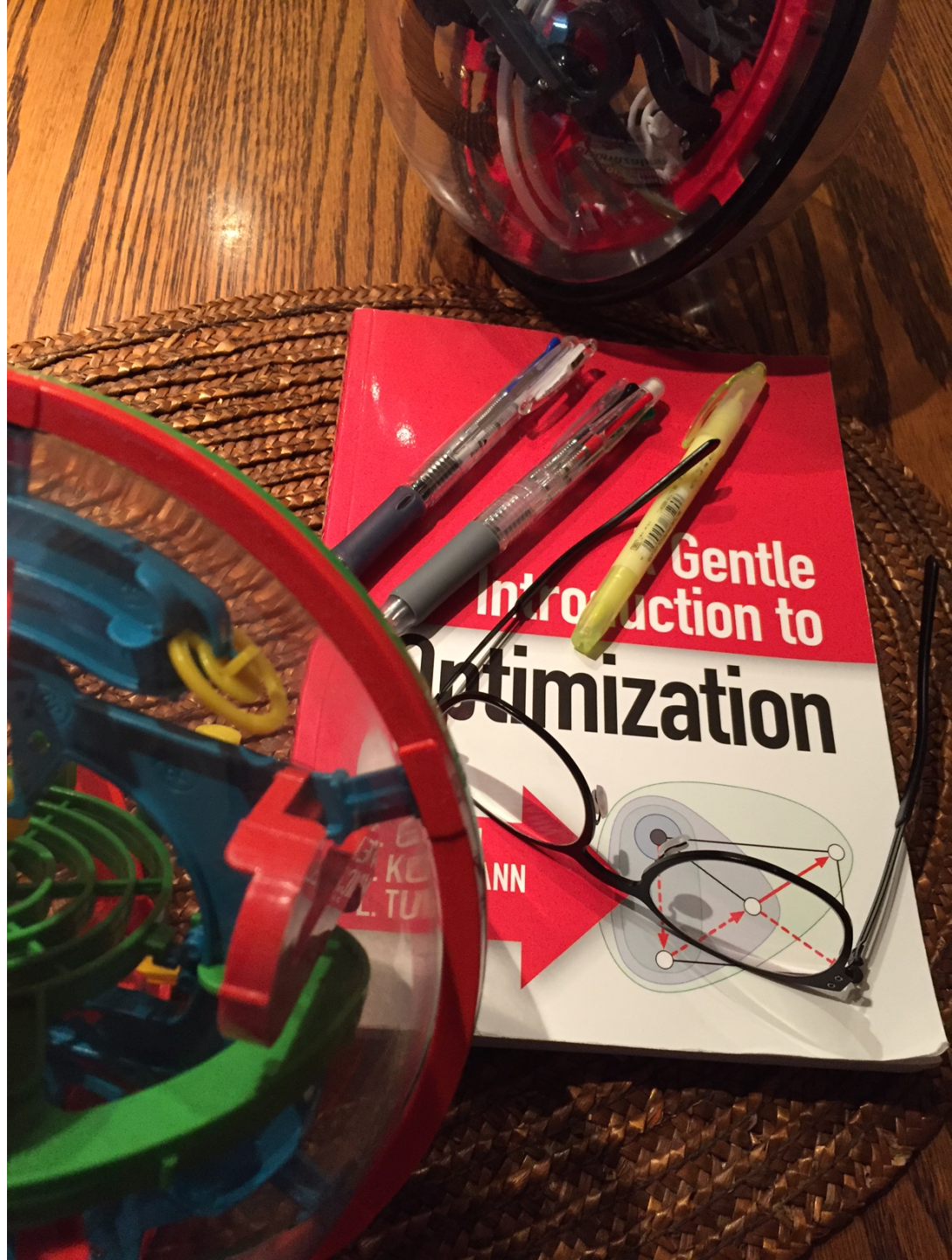


Optimization Theory

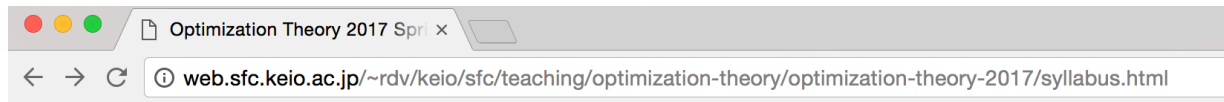
Assoc. Prof. Rodney
Van Meter

Assoc. Prof. Masashi Aono
Lecture 1



See the class web page!

<http://web.sfc.keio.ac.jp/~rdv/keio/sfc/teaching/optimization-theory/optimization-theory-2017/syllabus.html>



慶應義塾大学
2017年度 春学期

Optimization Theory

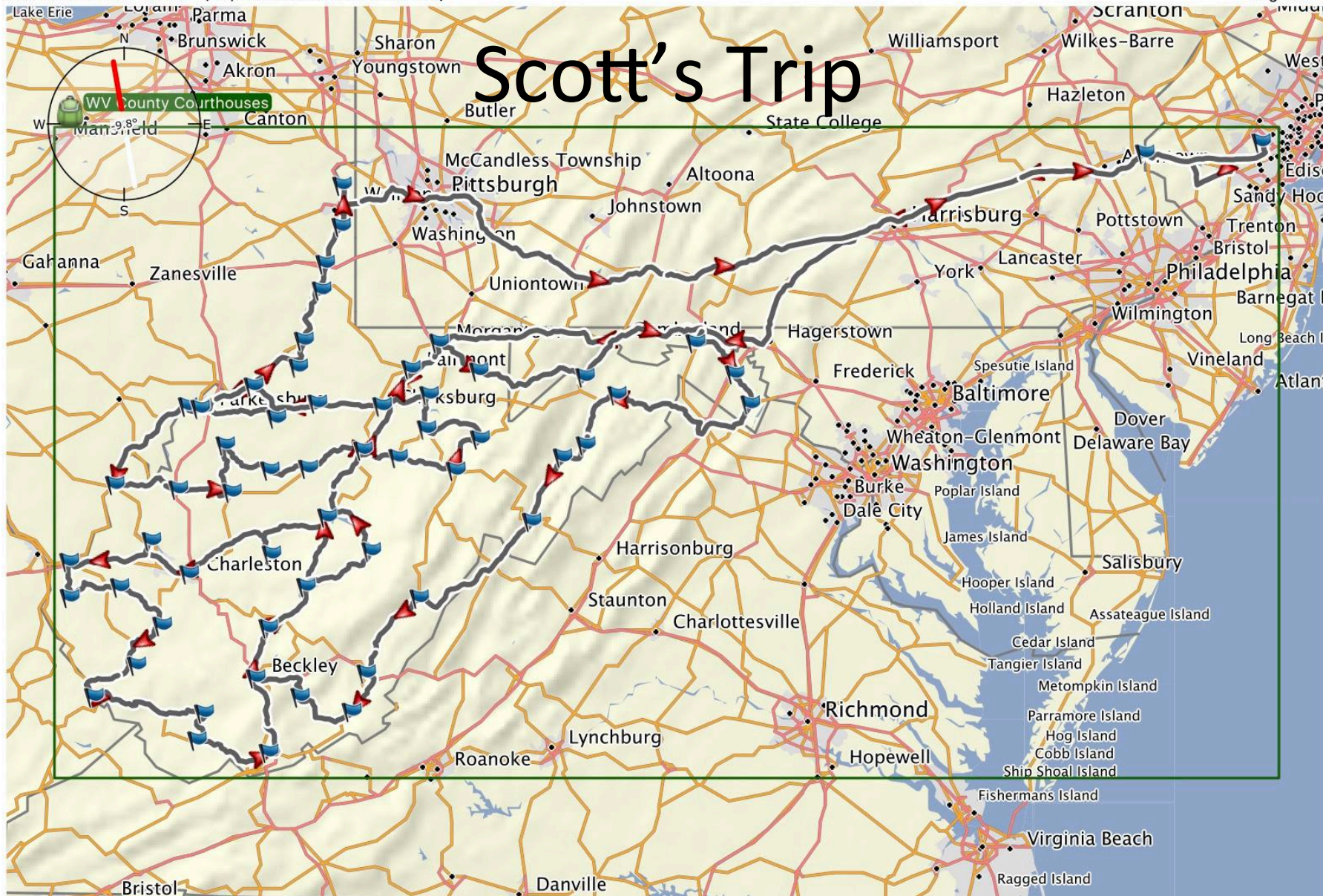
Note: To read the HTML files below properly, you will need a MathML-capable browser. This page is rendered using MathML. If you are using another browser, please use the PDF versions of the files.

2017年度春学期 水曜日 1 時限

2017 Spring Term, Class is Wednesdays 1st period.

担当: [Rodney Van Meter](#)

E-mail: rdv@sfc.keio.ac.jp



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

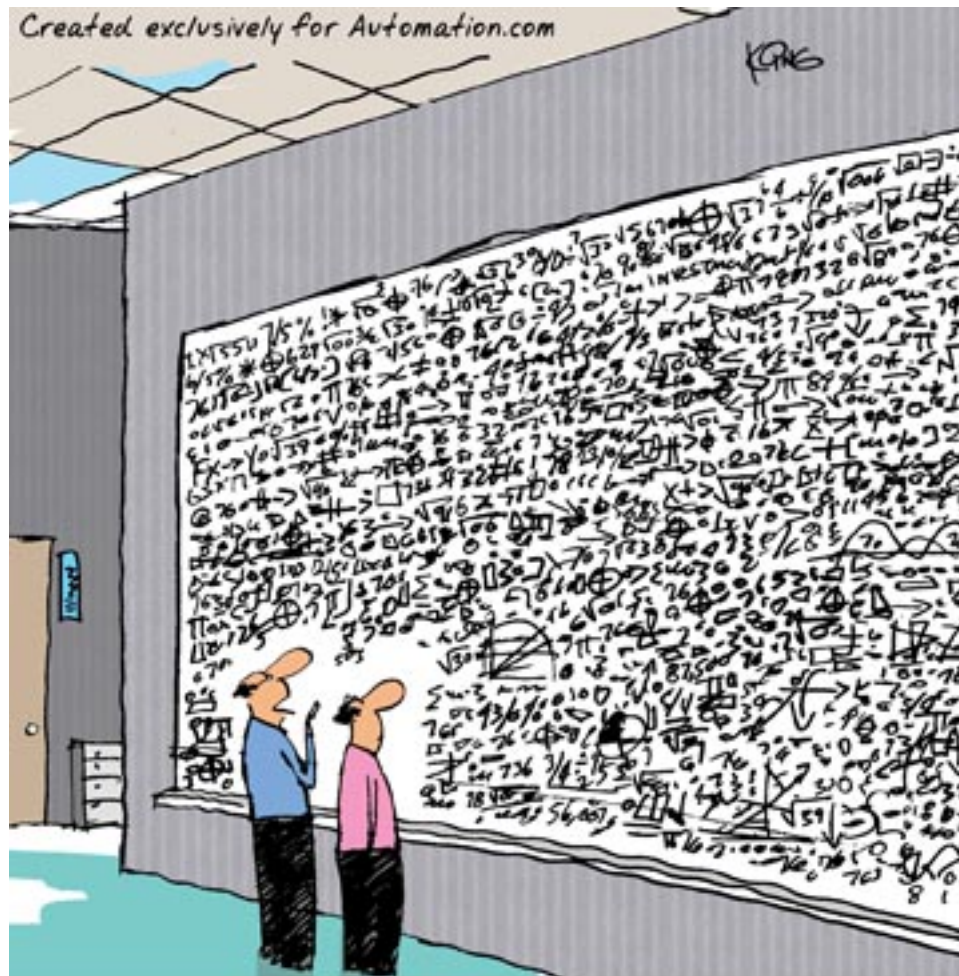


http://labs.blogs.com/its_alive_in_the_lab/2010/12/factory-layout-optimization-for-autocad-now-available.html



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

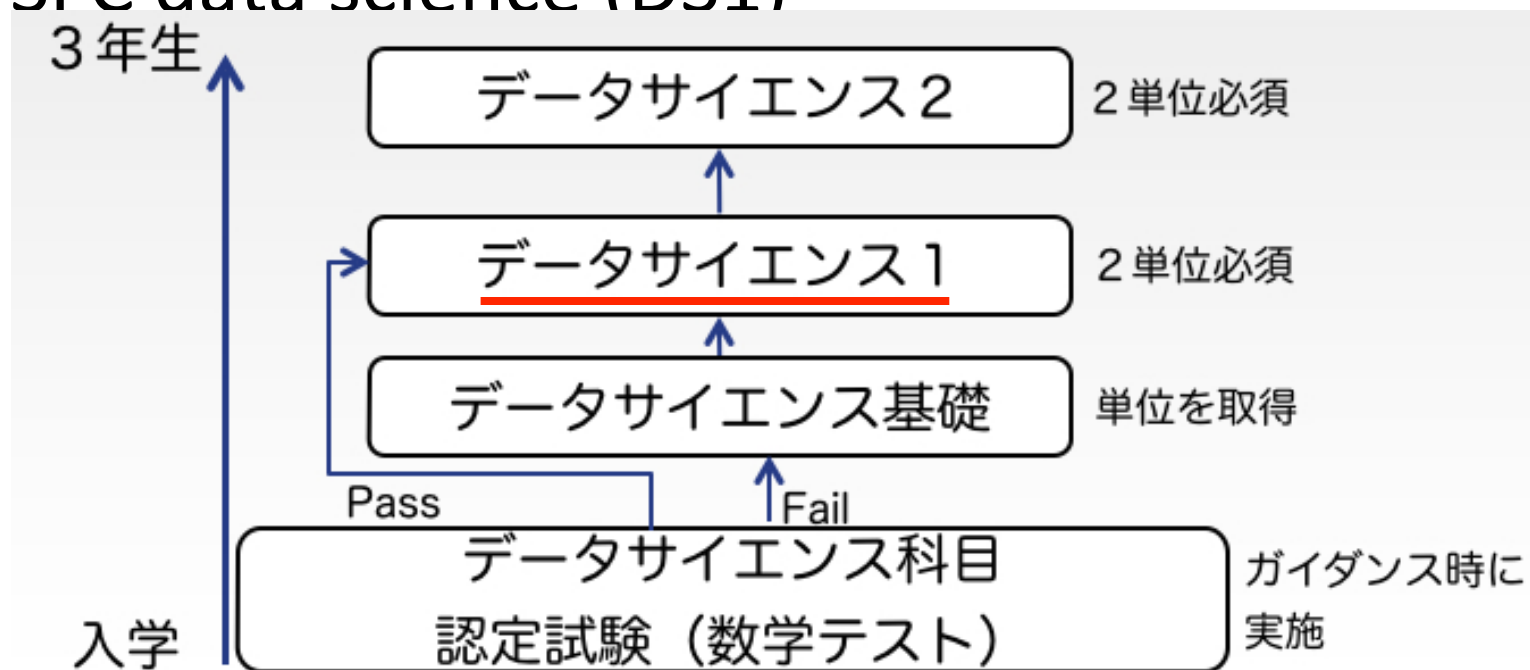
Factory Optimization

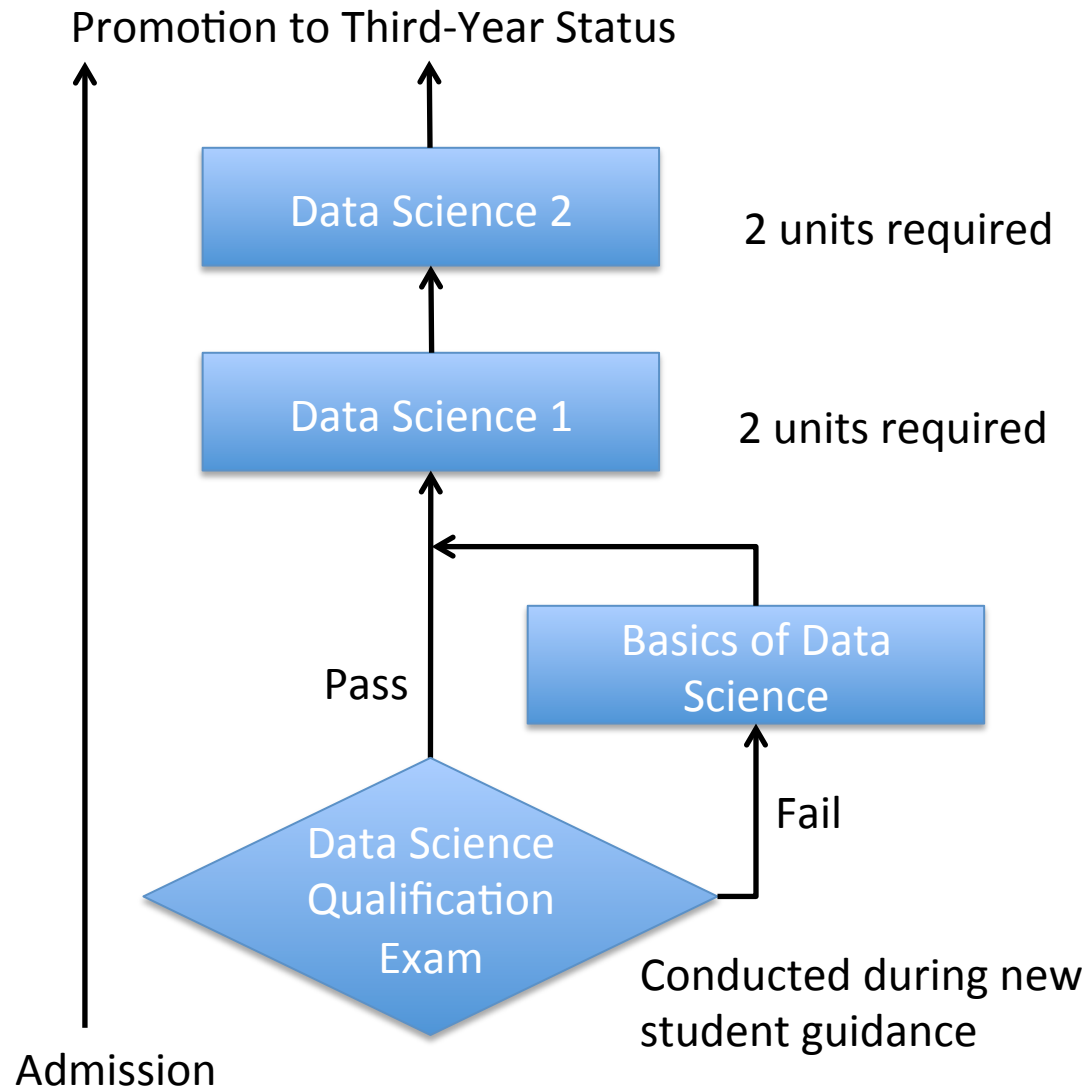


*“...and that, in simple terms, is my idea on
how to increase factory optimization.
any questions?”*

About this class in SFC curriculum

- SFC data science (DS1)





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

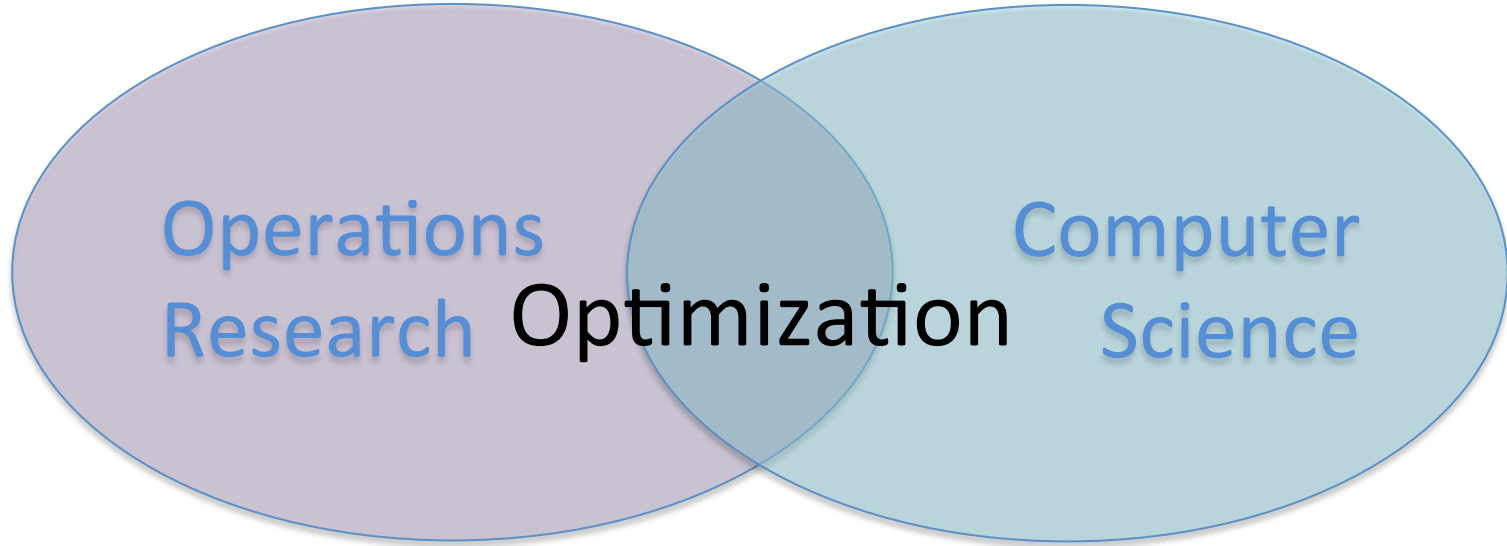
Student Backgrounds & Goals

- What are your goals for this class?
- How much math background do you have?
- How much programming experience do you have?

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?

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

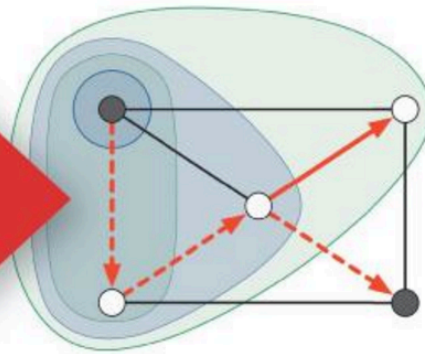
Same thing as LP, except that a non-empty subset of the variables must take integer values

- 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.

A Gentle Introduction to Optimization

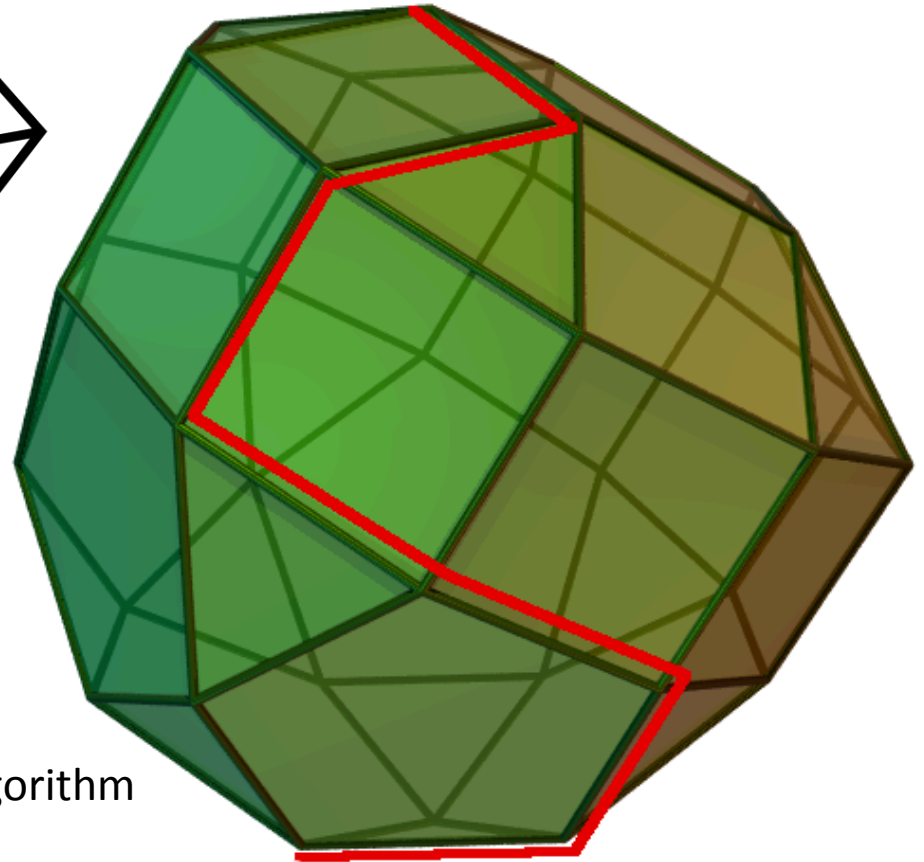
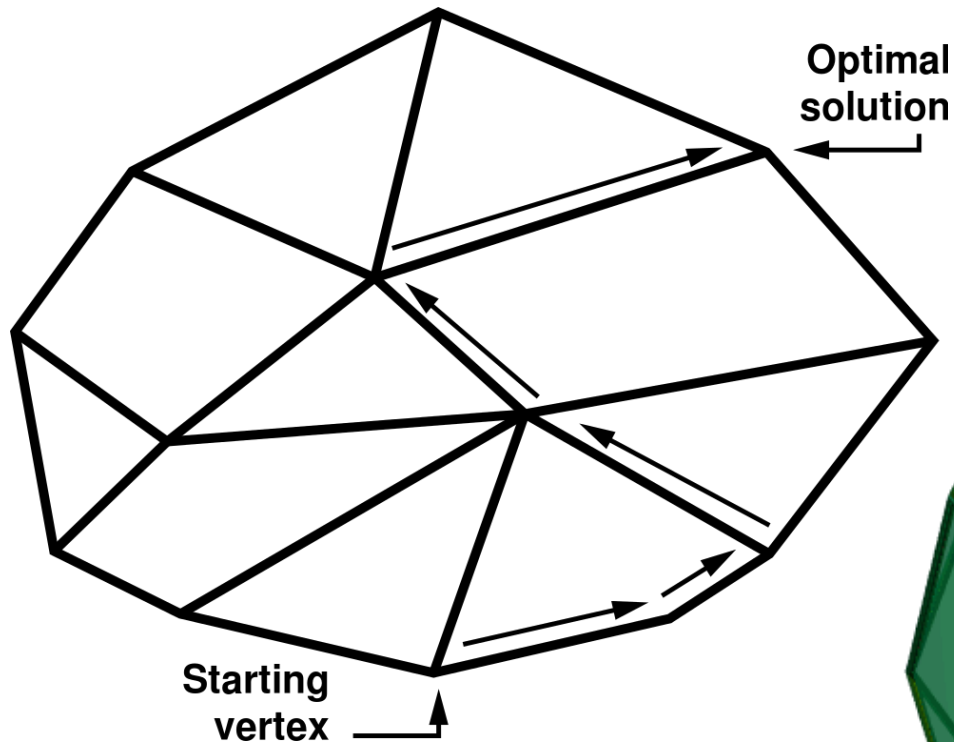
B. GUENIN
J. KÖNEMANN
L. TUNÇEL



This Semester

Week/ Chapter	Topic	Week/ Chapter	Topic
1/1	Introduction, Linear/Integer/ Nonlinear Problems, Business/ Manufacturing Examples	8/	Graph Algorithms 3: Distributed Algorithms
2/1.4-1.5	Formulating Formal Problems: on Graphs and in Networking	9/	Graph Algorithms 4: Minimum Cost Perfect Matching
3/2.1-2.4	Solving Linear Problems 1: Certificates	10/6.1-6.4	Integer Problems 1: Knapsack Problem
4/2.5-2.8	Solving Linear Problems 2: the Simplex Algorithm	11/App. A	Computational Complexity
5/3.1	Solving Linear Problems 3: the Simplex Algorithm, Complete	12/7	Nonlinear Optimization 1: Concepts
6/	Graph Algorithms 1: Concepts	13/none	Nonlinear Optimization 2: Algorithms
7/	Graph Algorithms 2: Basic Algorithms	14/none	Pareto Optimality, Sex and Wrapping Up the Semester

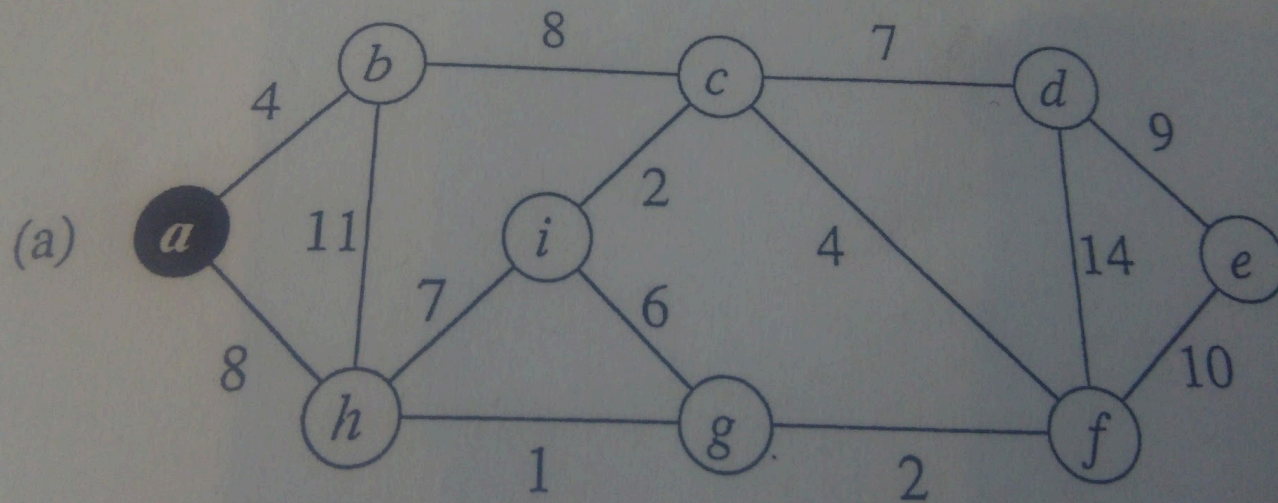
Linear Programs: Simplex Algorithm



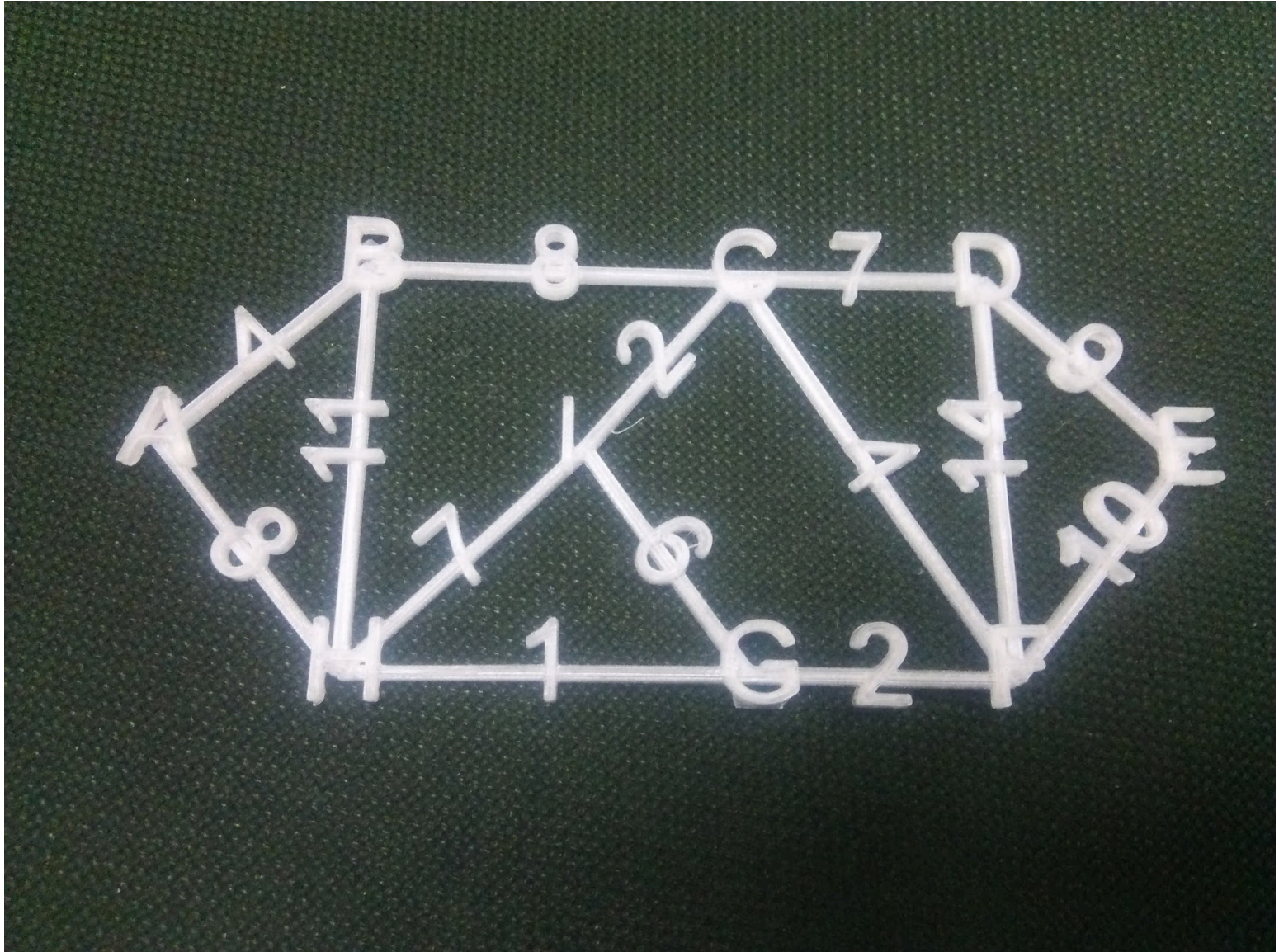
Images from
https://en.wikipedia.org/wiki/Simplex_algorithm

Graph Problems

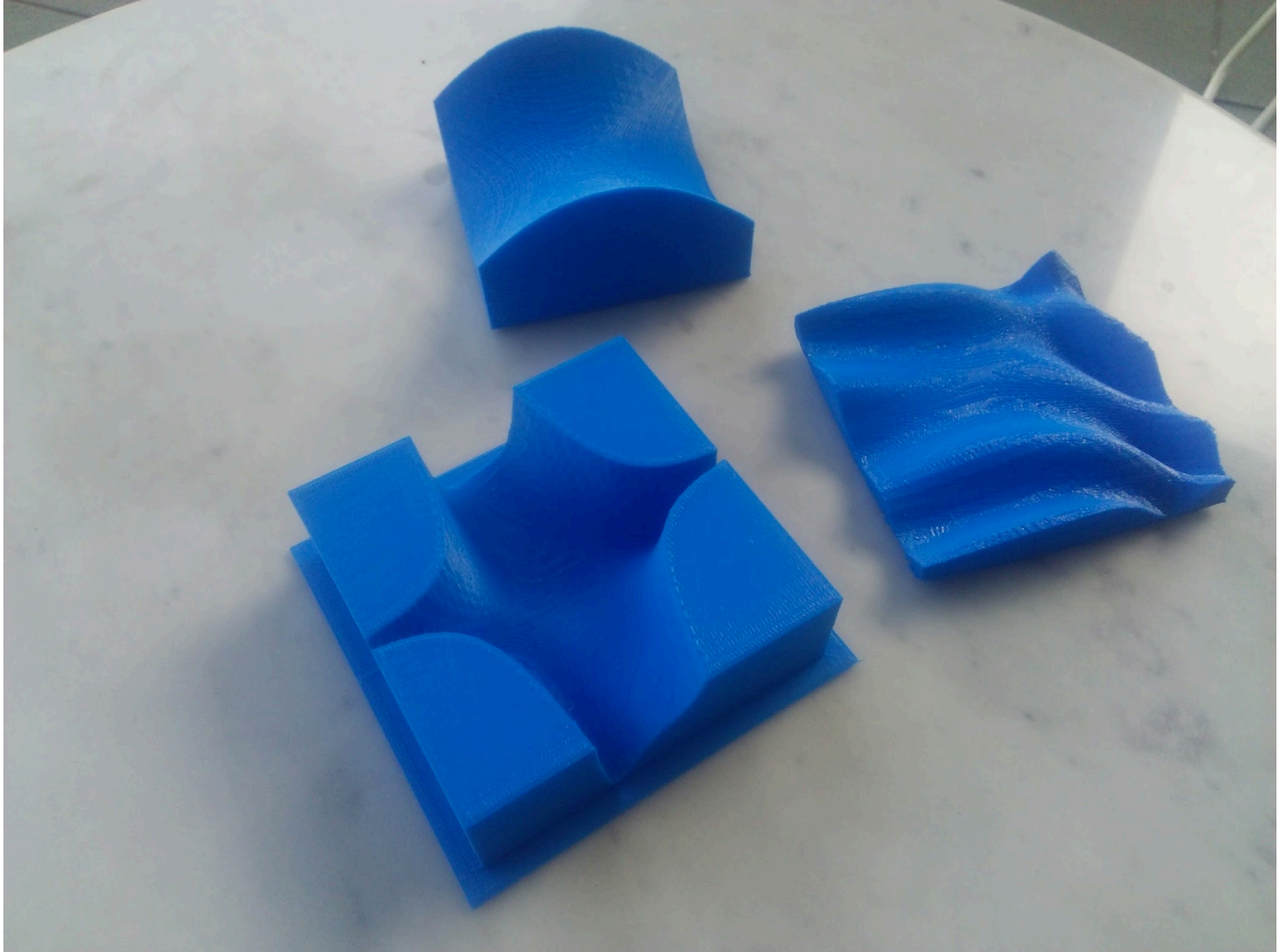
23.2 The algorithms of Kruskal and Prim



Graph Problems



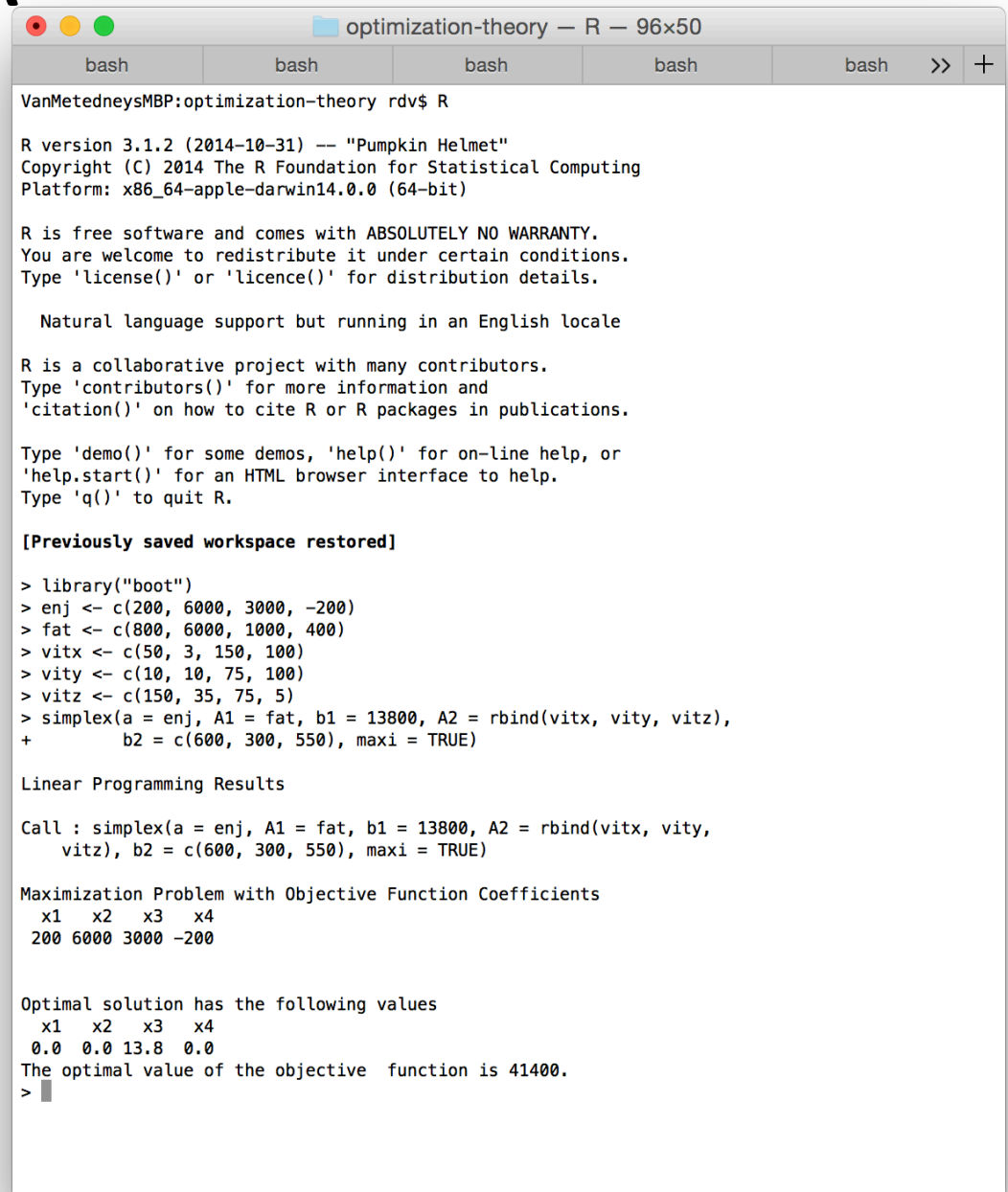
Non-linear Optimization



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 (submit via SFS, no presentation)
- 20% class participation (means we need to know your name!)

Homeworks

- 6 homeworks:
 - 1 on math background
 - 3 on simplex & linear programs
 - 2 on graph algorithms

Final Project

Examples:

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

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

Examples

- visualization of A* in JavaScript
- Dijkstra in Java
- Dijkstra in Haskell
- 3-D printed object (connection to optimization weak)
- long, thorough essay on Pareto
- created web app using a large optimization library, to let the user input a series of items to purchase and ship, and their constraints (weight, cost) and profit.
- Creation of a Wikipedia page in Bahasa Indonesian describing the Hungarian Algorithm
- creation of an efficient fantasy league baseball team using dynamic programming to solve the knapsack problem
- creating an advertising budget for a sports drink, for different kinds of media, with various constraints



Algoritma Hungaria

Dari Wikipedia bahasa Indonesia, ensiklopedia bebas

Metode Hongaria adalah algoritma optimasi kombinatorial yang menyelesaikan masalah untuk menyelesaikan soal yang berupa penugasan dengan cara menemukan pemasanga

Pada dasarnya, proses algorima ini melibatkan perubahan biaya di dalam array sampai be

Daftar isi [sembunyikan]

- 1 [Penjelasan yang mudah](#)
 - 1.1 [Langkah 1](#)
 - 1.2 [Langkah 2](#)
 - 1.3 [Langkah 3](#)
 - 1.4 [Langkah 4](#)
 - 1.5 [Langkah 5](#)
 - 1.6 [Langkah 6](#)
- 2 [Referensi](#)
- 3 [Bacaan lanjutan](#)

Penjelasan yang mudah [sunting | sunting sumber]

Dalam contoh ini, ada empat pekerja: Andre, Budi, Corry, dan Durian. Setiap pekerja bisa dan melap jendela. Tetapi, setiap pekerja meminta bayaran yang berbeda-beda untuk tug

Langkah 4 [sunting | sunting sumber]

Dengan menggunakan metode di atas, temukan angka minimal dalam se horizontal dan kolom 2 secara vertikal. Setiap barisan atau kolom itu dico

	1	2	3	4
A	25	0	10	1
B	5	50	0	0
C	5	0	10	15
D	0	0	5	5

Langkah 5 [sunting | sunting sumber]

Modifikasi semua entri di dalam matrix dengan cara mengurangi angka m anti-diagonal dan tambahkan nilai 1 untuk entri yang tidak tersentuh.

	1	2	3	4
A	24	0	9	0
B	5	51	0	0
C	4	0	9	14
D	0	1	5	5

Langkah 6 [sunting | sunting sumber]

Tetapkan pekerjaan berdasarkan angka minimal dari setiap baris.

	1	2	3	4
A	24	0	9	0
B	5	51	0	0
C	4	0	9	14
D	0	1	5	5

Dalam soal ini, total biaya minimal adalah Rp. 111,000 dengan penugasa