ORF 363/COS 323
Computing and Optimization in the Physical and Social Sciences

Amir Ali Ahmadi
Princeton, ORFE

Lecture 1
What is optimization?

- Roughly, can think of optimization as the science of making the most out of every situation.

- You’ve probably all done it many times this week:

  - **What courses to take?**
    - To maximize learning.
    - To maximize GPA (?!)
    - Courses can’t conflict.
    - Not before 10AM.
    - Professor rating > 4.5.

  - **What furniture to buy?**
    - To minimize cost.
    - To maximize comfort.
    - Must fit in your room.
    - Must have 3 drawers.
    - Not too heavy.

  - **Where to get dinner?**
    - To minimize cost.
    - Less than .5 miles from dormitory.
    - Must have ice cream for dessert.
    - Sanitation grade > 7.

- **Common theme:**
  - You make decisions and choose one of many alternatives.
  - You hope to maximize or minimize something (you have an objective).
  - You cannot make arbitrary decisions. Life puts constraints on you.
How is this class different from your every-day optimization?

- We’ll be learning techniques for dealing with problems that have
  - Thousands (if not millions) of variables
  - Thousands (if not millions) of constraints
- These problems appear every day in the industry, in science, in engineering
- Hopeless to make decisions in your head and with rules of thumb
- Need mathematical techniques that translate into algorithms
  - Algorithms then get implemented on a computer to solve your optimization problem
- We typically model a physical or social scenario with a precise mathematical description
- In this mathematical model, we care about actually finding the best solution
- Whenever we can’t find the best solution, we would like to know how far off our proposed solution is
Examples of optimization problems

**In finance**
- In what proportions to invest in 500 stocks?
  - To maximize return.
  - To minimize risk.
  - No more than 1/5 of your money in any one stock.
  - Transactions costs < $70.
  - Return rate > 2%.

**In machine learning**
- How to assign likelihoods to emails being spam?
  - To minimize probability of a false positive.
  - To penalize overfitting on training set.
  - Probability of false negative < .15.
  - Misclassification error on training set < 5%.

**In control engineering**
- How to drive an autonomous vehicle from A to B?
  - To minimize fuel consumption.
  - To minimize travel time.
  - Distance to closest obstacle > 2 meters.
  - Speed < 40 miles/hr.
  - Path needs to be smooth (no sudden changes in direction).
Computing and Optimization

▪ This class will give you a broad introduction to “optimization from a computational viewpoint.”

▪ Optimization and computing are very close areas of applied mathematics:
  ▪ For a host of major problems in computer science, the best algorithms currently come from the theory of optimization.
  ▪ Conversely, foundational work by computer scientists has led to a shift of focus in optimization theory from “mathematical analysis” to “computational mathematics.”

▪ Several basic topics in scientific computing (that we’ll cover in this course) are either special cases or fundamental ingredients of more elaborate optimization algorithms:
  ▪ Least squares, root finding, solving linear systems, solving linear inequalities, approximation and fitting, matrix factorizations, conjugate gradients,…
Agenda for today

▪ Meet your teaching staff
▪ Get your hands dirty with algorithms
  ▪ Game 1
  ▪ Game 2
▪ Modelling with a mathematical program
  ▪ Fermat’s last theorem!
▪ Course logistics and expectations
Meet your teaching staff (1/3)

- **Amir Ali Ahmadi** (Amir Ali, or Amirali, is my first name)

- I am an Assistant Professor at ORFE (since Fall 2014).

- I come here from MIT, EECS, after a fellowship at IBM Research.

- Office hours: **Tuesdays, 5:30-7:30 PM, Sherrerd 329.**
  
  *(Overflow room ➔ Sherrerd 125)*

http://aaa.princeton.edu/ a_a_a@p...
Meet your teaching staff (2/3)

- **Thomas Pumir**
  - Office hours: **Mon 4-6pm**, Sherrerd 005
  - tpumir@p...

- **Benjamin Mirabelli**
  - Office hours: **Tue 7:30-9:30pm**, Sherrerd 005
  - bpm2@p...

- **Cemil Dibek (honorary TA)**
  - Office hours: **Mon 6-8pm**, Sherrerd 005
  - cdibek@p...

- **Zheng Yu**
  - Office hours: **Wed 5-7pm**, Sherrerd 005
  - zhengy@p...

- **Francesca Tang**
  - Office hours: **Tue 9-11am**, Sherrerd 005
  - frtang@p...

- **Bachir El Khadir (honorary TA)**
  - Office hours: **Wed 7-9pm**, Sherrerd 005
  - bkhadir@p...
Meet your teaching staff (3/3)

- **Yaqi Duan**
  - Office hours: **Thu 9-11am, Sherrerd 006**
  - yaqid@p...

- **Wenyang Gong**
  - Office hours: **Fri 3-5pm, Sherrerd 005**
  - wenyang@p...

- **Jeffrey Zhang (honorary TA)**
  - Piazza Guru
  - jeffz@p...
Meet your classmates

ORF 363/COS 323, Fall 2018 (110 registered students)
Let’s get to the games!
Let’s ship some oil together!

- **Rules of the game:**
  - Cannot exceed capacity on the edges.
  - For each node, except for S and T, flow in = flow out (i.e., no storage).
- **Goal:** ship as much oil as you can from S to T.
Let me start things off for you. Here is a flow with value 10:

Can you do better? How much better?

You all get a copy of this graph on the handout.

You have 7 minutes!
You tell me, I draw...
A couple of good attempts

- Flow of value 15
- Can you do better?

- Flow of value 17
- Can you do better?
- How can you prove that it’s impossible to do better?
17 is the best possible!

- Proof by magic:
  - The rabbit is the red “cut”!
  - Any flow from S to T must cross the red curve.
  - So it can have value at most $2+2+3+3+4+2+1=17$.

- And here is the magic: such a proof is always possible!
Let’s try a more realistic graph

- You have 7 minutes! ;)}
▪ How long do you think an optimization solver would take (on my laptop) to find the best solution here?
▪ How many lines of code do you think you have to write for it?
▪ How would someone who hasn’t seen optimization approach this?
  ▪ Trial and error?
  ▪ Push a little flow here, a little there...
  ▪ Do you think they are likely to find the best solution?
A bit of history behind this map

- From a secret report by Harris and Ross (1955) written for the Air Force.
- Railway network of the Western Soviet Union going to Eastern Europe.
- Declassified in 1999.
- Look at the min-cut on the map (called the “bottleneck”)! There are 44 vertices, 105 edges, and the max flow is 163K.
- Harris and Ross gave a heuristic which happened to solve the problem optimally in this case.
- Later that year (1955), the famous Ford-Fulkerson algorithm came out of the RAND corporation. The algorithm always finds the best solution (for rational edge costs).

More on this history: [Sch05]
Let’s look at a second problem

...and tell me which one you thought was easier
Two finals in one day? No thanks.

- The department chair at ORFE would like to schedule the final exams for 14 undergraduate courses offered this semester.

- He wants to have as many exams as possible on the same day, so everyone gets done quickly and goes on vacation.

- There is just one constraint: No student should have >1 exam on the same day.

- The nodes of this graph are the 14 courses.

- There is an edge between two nodes if and only if there is at least one student who is taking both courses.

- If we want to schedule as many exams as possible on the same day, what are we looking for in this graph?

  - The largest collection of nodes such that no two nodes share an edge.
Let me start things off for you. Here is 4 concurrent final exams:

- Can you do better?
- How much better?
- You all get a copy of this graph on the handout.

You have 7 minutes!
You tell me, I draw...
A good attempt

- Tired of trying?
- Is this the best possible?

6 exams
6 is the best possible!

▪ Proof by magic?
  ▪ Unfortunately not 😞
  ▪ No magician in the world has pulled out such a rabbit to this day! (By this we mean a trick that would work on all graphs.)

▪ Of course there is always a proof:
  ▪ Try all possible subsets of 7 nodes.
  ▪ There are 3432 of them.
  ▪ Observe that none of them work.

▪ But this is no magic. It impresses nobody. We want a “short” proof. (We will formalize what this means.) Like the one in our max-flow example.

▪ Let’s appreciate this further...
Let’s try another graph

- Encouraged by the success of ORFE, now the Dean of Engineering wants to do the same for 115 SEAS courses.

- How many final exams on the same day are possible? Can you do 17?
  - You have 7 minutes! ;)

- Want to try out all possibilities for 17 exams?
  - There are over 80000000000000000000 of them!
But there is some good news

- Even though finding the best solution always may be too much to hope for, techniques from optimization (and in particular from the area of *convex optimization*) often allow us to find high-quality solutions with performance guarantees.

- For example, an optimization algorithm may quickly find 16 concurrent exams for you.

- You really want to know if 17 is impossible. Instead, another optimization algorithm (or sometimes the same one) tells you that 19 is impossible.

- This is very useful information! You know you got 16, and no one can do better than 19.

- *We sill see a lot of convex optimization in this class!*
Which of the two problems was harder for you?

- Not always obvious. A lot of research in optimization and computer science goes into distinguishing the “tractable” problems from the “intractable” ones.

- The two brain teasers actually just gave you a taste of the P vs. NP problem. (If you have not heard about this, that’s OK. You will soon.)

- The first problem we can solve efficiently (in “polynomial time”).

- The second problem: no one knows. If you do, you literally get $1M!
  
  More importantly, your algorithm immediately translates to an efficient algorithm for thousands of other problems no one knows how to solve.
Modelling problems as a mathematical program
Let’s revisit our first game

- **Rules of the game:**
  - Cannot exceed capacity on the edges.
  - For each node, except for S and T, flow in = flow out (i.e., no storage).
  - **Goal:** ship as much oil as you can from S to T.

- What were your decision variables?
- What were your constraints?
- What was your objective function?
\[ \kappa_{SA}, \kappa_{AD}, \kappa_{BE}, \ldots, \kappa_{GT} \]

Decision variables

\[ \max. \quad \kappa_{SA} + \kappa_{SB} + \kappa_{SC} \]

Objective function

s.t.

- \[ \kappa_{SA}, \kappa_{AD}, \kappa_{BE}, \ldots, \kappa_{GT} \geq 0 \]

- \[ \kappa_{SA} \leq 6, \kappa_{AB} \leq 2, \kappa_{EG} \leq 10, \ldots, \kappa_{GT} \leq 12 \]

Constraints

\[
\begin{align*}
\kappa_{SA} &= \kappa_{AD} + \kappa_{AB} + \kappa_{AE} \\
\kappa_{SC} &= \kappa_{CB} + \kappa_{CF} \\
&\vdots \\
\kappa_{CF} + \kappa_{EF} &= \kappa_{FT}.
\end{align*}
\]
Let’s revisit our second game

- What were your decision variables?
- What were your constraints?
- What was your objective function?
$\lambda_1, \lambda_2, \ldots, \lambda_{12}$  

Objective function

$s.t.$

0  

Constraints

\begin{align*}
\lambda_i (1 - \lambda_i) &= 0, \quad i = 1, \ldots, 12 \\
\lambda_1 + \lambda_2 &\leq 1 \\
\lambda_1 + \lambda_8 &\leq 1 \\
\lambda_4 + \lambda_6 &\leq 1 \\
\vdots \\
\lambda_{12} + \lambda_8 &\leq 1
\end{align*}
Why one hard and one easy? How can you tell?

\[ \begin{align*}
\mathcal{K}_{SA}, & \quad \mathcal{K}_{AD}, \quad \mathcal{K}_{BE}, \ldots, \quad \mathcal{K}_{GT} \\
\max. & \quad \mathcal{K}_{SA} + \mathcal{K}_{SB} + \mathcal{K}_{SC} \\
\text{s.t.} & \\
0 & \quad \mathcal{K}_{SA}, \quad \mathcal{K}_{AD}, \quad \mathcal{K}_{BE}, \ldots, \quad \mathcal{K}_{GT} \geq 0 \\
0 & \quad \mathcal{K}_{SA} \leq 6, \quad \mathcal{K}_{AB} \leq 2, \quad \mathcal{K}_{EG} \leq 10, \ldots, \quad \mathcal{K}_{GT} \leq 12 \\
0 & \quad \mathcal{K}_{SA} = \mathcal{K}_{AD} + \mathcal{K}_{AB} + \mathcal{K}_{AE} \\
0 & \quad \mathcal{K}_{SC} = \mathcal{K}_{CB} + \mathcal{K}_{CF} \\
0 & \quad \mathcal{K}_{CF} + \mathcal{K}_{EF} = \mathcal{K}_{FF}. 
\end{align*} \]

\[ \begin{align*}
\mathcal{X}_1, \mathcal{X}_2, \ldots, & \quad \mathcal{X}_{12} \\
\max. & \quad \mathcal{X}_1 + \mathcal{X}_2 + \ldots + \mathcal{X}_{12} \\
\text{s.t.} & \\
0 & \quad \mathcal{X}_i (1 - \mathcal{X}_i) = 0, \quad i = 1, \ldots, 12 \\
0 & \quad \mathcal{X}_1 + \mathcal{X}_2 \leq 1 \\
0 & \quad \mathcal{X}_4 + \mathcal{X}_6 \leq 1 \\
0 & \quad \mathcal{X}_{12} + \mathcal{X}_8 \leq 1 \\
\end{align*} \]

**Caution:** just because we can write something as a mathematical program, it doesn’t mean we can solve it.
Fermat’s Last Theorem

Can you give me three positive integers $x, y, z$ such that

$$x^2 + y^2 = z^2?$$

Sure: \[(3, 4, 5)\] \[(5, 12, 13)\] \[(8, 15, 17)\] \[(7, 24, 25)\] \[(20, 21, 29)\] \[(12, 35, 37)\] \[(9, 40, 41)\] \[(28, 45, 53)\]

And there are infinitely many more...

How about \[x^3 + y^3 = z^3?\]

How about \[x^4 + y^4 = z^4?\]

How about \[x^5 + y^5 = z^5?\]
Fermat’s Last Theorem

Fermat’s conjecture (1637):
For $n \geq 3$, the equation $x^n + y^n = z^n$ has no solution over positive integers.

Proved in 1994 (357 years later!) by Andrew Wiles. (Was on the faculty in our math department until a few years ago.)
Fermat’s Last Theorem

- Fermat’s conjecture (1637):
  For $n \geq 3$, the equation $x^n + y^n = z^n$ has no solution over positive integers.

- Consider the following optimization problem (mathematical program):

  $$\min_{x,y,z,n} \left( (x^n + y^n - z^n)^2 \right)$$

  s.t. 
  $$x, y, z, n \geq 1, \quad n \geq 3,$$

  $$\sin^2(\pi n) + \sin^2(\pi x) + \sin^2(\pi y) + \sin^2(\pi z) = 0.$$
Course objectives

- The skills I hope you acquire:

- Ability to view your own field through the lens of optimization and computation
  - To help you, we’ll draw applications from operations research, statistics, finance, machine learning, engineering, ...

- Learn about several topics in scientific computing

- More mathematical maturity and ability for rigorous reasoning
  - There will be some proofs in lecture. Easier ones on homework.

- Enhance your coding abilities (nothing too fancy, simple MATLAB)
  - There will be a MATLAB component on every homework and on the take-home final.

- Ability to recognize hard and easy optimization problems

- Ability to use optimization software
  - Understand the algorithms behind the software for some easier subclass of problems.
Things you need to download

- Right away:
  
  MATLAB

  http://www.princeton.edu/software/licenses/software/matlab/

- In the next week or two (will appear on HW#2 or #3):
  
  CVX

  http://cvxr.com/cvx/
Course logistics

- On blackboard (and will be on Blackboard).
- Course website:
  
  http://aaa.princeton.edu/orf363

- For those interested:
  
  - Princeton Optimization Seminar (Thursdays 4:30 PM)
  - http://orfe.princeton.edu/events/optimization-seminar

Image credits and references:
