Course website:
chemaoxfz.github.io/opt_app

CST 5020 Optimization and Applications

Fangzhou Xiao, Westlake Univ.
肖方舟，西湖大学
2025 SP

Course: motivation

What this course is about:

- optimization, how to write down a problem, how to solve one, how to think about it, and its applications.

Why I'm teaching:

- I'm a bioengineer, synthetic biologist. I encounter optimization frequently when analyze and design **complex** biological systems. I learned it here and there, but I want to learn it systematically.

Why you'd want to learn:

- For scientists: optimization is (still) an under-appreciated way to specify and understand complex systems, esp. in biology.
- For engineers: you already encounter optimization a lot, it's worth it to know more about what is going on inside the algorithms and problem formulations.

Course: the environment and the spirit

Background needed: I'll adapt!

- It is helpful if you're familiar with linear algebra, real analysis, and some coding. But you can pick them all up along the way. Just put in the effort and it will be worth it. I can adapt the pace of the course accordingly.

Diverse student composition:

- Diverse background and goals: biology, physics, bio/comp engineering.
- So there will be some topics that you appreciate and understand, and some that you don't. It's fine. Ask questions whenever it's bewildering!
- Think of it as you're gaining multiple "hats" that you can put on.

Spirit: we're learning about this awesome topic together!

- Think of me as a peer. I'm literally only <20 hours of reading ahead of you.
- Just like there will be concepts you don't understand, so will I. Let's discuss!

Course: Problem sets and grading

Problem sets:

- There will be roughly 1 problem set every 2 weeks.
    - Takes ~20 hours each, so start early!
- Hand them in before the next pset is handed out.
- I'll collect but I won't grade on correctness, but grade on **effort**.
- Write in latex, upload in one pdf. (overleaf can be helpful.) Upload on canvas.

Grading

- Problem sets 10% x 8
- Active participation 10%
- Final take-home exam 10%

You can always choose to not hand in one problem set (or the final exam) and get 10% for free.

Collaboration: You are encouraged to work in small groups, or talk to AI, but please write up your solutions independently and make sure you understand what you write.

My take on "Why Optimization"?

- Optimization is not just a method or a tool, it's a "Language" that expresses how we understand and engineer the complex world.
    - The natural language for citizens of an **industrialized** civilization.
- An anecdotal history of how we interact with the world:
    - Pre-industrialization - obey, observe. A **descriptive** language.
        - Math: numbers and functions.
        - Complexity as society develops. Such math can't describe complex objects or behaviors.
        - Trajectory vs energy min / entropy max
        - Newton's laws vs Lagrangian - principle of least action.
    - 18-19th century: to simply the complexity without reduction, change perspective to optimize!
        - Many observations to describe, the best one?
            - Gauss, least squares
        - Many things we can do, "knobs" we can tune, for a "goal"!
            - Constraints and objectives, e.g. optimal transport

My take on "Why Optimization"? Describe vs Optimize

- Describing:
  - A point, specific.
  - "You get what you see", easy to use (calc) once you have it.
  - Can't describe complex objects or behaviors.
  - Have to simplify by reduction. e.g. minimal models, hydrogen atom.
    - Complexity in objects are a curse….
    - Allowing more behaviors -- increase in complexity of description.
- Optimizing:
  - A set, non-specific, capturing all points in the constraint.
  - Needs some **computation** to "solve".
  - Can capture complex objects and behaviors.
  - Simplify without reduction.
    - Complexity in objects are a bless!!!
    - Allowing more behaviors -- decrease in complexity of optimization.

The power of optimization as a language evolves with computational capacity

- Optimization needs computation to solve.
  - If a type of problems can be **routinely** solved, then it is part of our vocabulary -- once you specified in this form, then we're done.
- The problems that can be routinely solved evolve over time.
  - Pre 1600s: Numbers and shapes
  - 16-1700s: Elementary functions
  - 17 - 1800s: Least squares
  - 1900s: Linear programming
  - 1990s - today: Convex optimization
  - Future: Deep learning? Combinatorial optimization? Quantum comp?
- Some examples to give a perspective.
- "**Solve by Recognizing**." recognize convexity is often very profound…

What do we do in this course?

- We learn together.
- 1st half is on convex optimization basics.
  - Follow Stephen Boyd's book and his course.
  - The book is freely available online.
  - https://web.stanford.edu/class/ee364a/lectures.html
- 2nd half is on applications to specific domains.
  - Machine learning, control, bioengineering, finance.
  - Topics could vary depending on interest.