

Day 4: Optimization
Quantitative Engineering Analysis
Spring 2019

1 Schedule

- 0900-0945: Debrief and Demos
- 0945-1030: Intro to Optimization
- 1030-1045: Coffee
- 1045-1145: The World of Optimization
- 1145-1230: Optimization Framing

2 Debrief/Demos on the Bridge of Doom [45 mins]

- With another student at your table, draw a concept map of the main ideas you used in this challenge. Try to make connections to ideas from the previous two modules. Please take a picture/scan the concept map and upload it at the end of class.
- With your table mates, discuss what you tried, what your results were, what is still unknown, and how well do you think you characterized the model's accuracy. Please document this on the white boards. We will ask you to report out on this to the whole class.
- With your table mates, discuss your process in the Bridge of Doom challenge. What worked and what did not as you engaged with this challenge? What would you do differently next time? We will ask you to report out on this to the whole class.
- Please let the teaching team know if you would like to do a live demo of your robot driving the Bridge of Doom™, or if you would like to show your video.

3 Overview

Over the course of the next few days we'll be learning about and applying ideas around optimization to the motion of a NEATO. You might have dealt with some of this before – e.g., maximum or minimum problems in calculus. We'll be broadening these ideas to deal with functions of many variables, as well as applying linear algebra ideas in order to allow us to optimize in situations that are much more complex than those typically found in a calculus textbook.

Optimization is really a type of modeling work: just as in ModSim, you need to abstract your physical system to a model, and then you need to apply mathematics and computational ideas in order to make a prediction with that model.

The abstraction component of optimization is an important one, both from an intellectual perspective and from a practical perspective. Intellectually, a great deal of the work in doing meaningful optimization is figuring out what you're going to include in and exclude from your model, and in deciding how you can turn what are often vague implicit mental models into more formal (and computable) mathematical models.

From a practical perspective, there is a pretty standard way that optimization problems are framed, and knowing how to set up an optimization problem according to these standards is really important both so that you can use existing optimization tools (e.g., in MATLAB) as well as so that you can evaluate *which* tools might be more appropriate for the problem you're facing.

4 The Language of Optimization [45 mins]

Many optimization problems boil down to this question: given a set of *constraints*, what combination of *decision variables* leads to the best possible value of the *objective function*?

4.1 Objective Function

The objective function is the quantity that you are trying to maximize or minimize. Maybe you're trying to minimize the cost of a product by choosing appropriate materials and manufacturing approaches; maybe you're trying to maximize the profit of a business by choosing what quantities of what products to stock; maybe you're trying to maximize the fatigue life of a joint. In each of these cases, there is a clearly-defined quantity (cost, profit, fatigue life) that you are trying to optimize. And in each of these cases, you would need a model for that quantity – a way of predicting what value that quantity would

have for a given design.

4.2 Decision Variables

Decision variables are basically the “knobs” you can turn to optimize the objective function. Sometimes decision variables are continuous quantities (e.g., how long a member in a truss is) and other times they must take on discrete values (e.g., the number of cases of Twinkies you order for your business). Your objective function is a model which takes the decision variable as inputs, and outputs the quantity you’re trying to optimize.

4.3 Constraints

The thing that often makes optimization hard is the fact that you don’t have unlimited choice when you “turn the knobs”. For example, while you can make a truss as strong as you want to by adding material, presumably doing so increases both the cost and the mass of the truss – and both of these might be constrained.

Often constraints involve inequalities: you might want to maximize the strength of the truss subject to the constraint that the mass is less than 100 kg and the cost is less than 200 dollars.

Like objective functions, constraints require mathematical models that predict the constrained quantity as a function of the decision variables.

4.4 An Example

Let’s say you were trying to build an optimization model for investing. You have a certain number of dollars available to invest, a set of n different stocks to choose from; each stock has an expected return of r_i and an expected risk σ_i .

You might say “I’d like to get the highest profit possible, while keeping the risk below some acceptable level”. To state this mathematically, we’d do something like this:

Let w_i represent the fraction of the portfolio invested in stock i . Then the expected return for the portfolio is given by

$$E(\mathbf{w}) = \sum_i w_i r_i$$

and the risk associated with the portfolio is given by

$$\sigma_{total} = \sqrt{\sum_i w_i^2 \sigma_i^2}$$

Our decision variables are the investments that we make, \mathbf{w} . Our objective function is the expected profit, E . Our constraints are that

the size of the portfolio is bounded by the money we have to invest, and the risk is less than or equal to the acceptable level. Mathematically, we'd state the problem like this:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{maximize}} && \sum_i r_i w_i \\ & \text{subject to} && \sum_i w_i \leq 1, \\ & && \sqrt{\sum_i w_i^2 \sigma_i^2} \leq \sigma_{\max} \end{aligned}$$

which we would read as follows: find the vector \mathbf{w} that maximizes the expected profit subject to the total investments being less than or equal to the money we have to invest and subject to the risk being less than or equal to a specified amount.

4.5 Solving an optimization problem - or trying to...

For the portfolio problem above, please *perform* (or attempt) the optimization – i.e., find the portfolio distribution to maximize profit for a given level of acceptable risk. You choose the risk level.

1. For a two stock portfolio, with $r_1 = 0.1$, $r_2 = 0.2$, $\sigma_1 = 0.1$, $\sigma_2 = 0.05$,
2. For a two stock portfolio, with $r_1 = 0.1$, $r_2 = 0.2$, $\sigma_1 = 0.1$, $\sigma_2 = 0.3$,
3. For an eight stock portfolio, where $\mathbf{r} = [0.1, 0.13, 0.15, 0.17, 0.2, 0.23, 0.25, 0.28]$ and $\sigma = [0.2, 0.1, 0.4, 0.25, 0.1, 0.35, 0.32, 0.35]$.

4.6 Optimization in Smile or Facial Recognition

For the two problems below, please propose the appropriate formal mathematical framing for the optimization that you already performed. Note that you could also formulate the boat this way, but it's a bit more messy, so we're letting you off the hook on that one!

4. Given a set of training data of images which have been tagged as smiling or not, formulate the design of a smile-predictor algorithm as an optimization problem.
5. Given a set of training data of images which have been tagged as individuals, formulate an eigen-face facial recognition algorithm as an optimization problem.

5 Coffee [15 minutes]

6 The World of Optimization [1 Hr]

The world of optimization is large and rambling, and we'd like you to get a handle of the layout of this land. A really good place to start is the [NEOS Guide](#) from the Network-Enabled Optimization System. To help with your review, we suggest that you take the following steps:

6. Please spend some time reviewing the **Optimization Guide** at this site. It includes some basic definitions, and a comprehensive list and description of different types of optimization problems. It also includes a listing of algorithms used for optimization problems, but these are not well-organized so may be very confusing - don't spend too much time here unless you really want to. There is also a taxonomy but it is by no means complete. At the very least, you should be familiar with the following concepts:
 - (a) Continuous versus Discrete
 - (b) Unconstrained versus Constrained
 - (c) Linear versus Nonlinear
 - (d) Number of Objective Functions
 - (e) Deterministic versus Stochastic
 - (f) Linear Programming
 - (g) Unconstrained Optimization
7. Once you've reviewed the **Optimization Guide**, please spend some time reviewing the **Case Studies**. There are lots to choose from, but you will probably find the greatest number in the section called **Supply Chain**. In each case study you will find a problem statement or description, as well as a mathematical formulation of the optimization problem. At the end of each case study you will find details of the implementation on the NEOS server - just ignore this part entirely.
 - (a) Pick at least two of these case studies, review them, and read the material from the guide on that type of optimization problem.

7 Optimization Framing: Real World Cases [45 min]

The three cases below describe a real-world situation that can be framed as an optimization problem. Starting with the hopper problem, we would like you to formally restate these problems as a mathematical optimization problem. Note that this will require you to decide what your decision variables are, and how to represent your constraints and objective function mathematically (or experimentally). Also note that this is modeling work: you'll need to iterate, and you'll need to decide what to ignore. When you have a model that you are happy with, use the NEOS guide to classify the type of problem that you have defined, e.g. linear programming, mixed-integer nonlinear programming.

Once you've done the hopper problem, choose one of the other two, and formally frame that problem as an optimization problem too.

When you are done, prepare your work on the board, appropriate for review by other members of the class.

8. Parametric Hopper Design

Consider an "A Frame" hopper design. You'd like to make the hopper jump as high as possible – without breaking the DesNat rules. How should you design the hopper?

9. Racial Balancing

In 1968, the Supreme Court ruled that all school systems should desegregate immediately – but while the court specified that desegregation must occur, it did not specify what constituted an acceptable racial balance, or what approaches should be used to achieve this balance. Forced busing was adopted by many school districts. Propose a busing plan for a given school district, and formulate it as an optimization problem.

10. Gerrymandering

As you're probably aware, there is an enormous amount of discussion of how to draw lines defining voting districts fairly (or unfairly, as the case may be). Propose an approach for drawing district maps.