

Lecture 1

Introduction

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)*

1 Mathematical optimization, and this course

One cannot overstate how pervasive and far reaching mathematical optimization theory is. Optimization is everywhere. This is true in two senses:

- Most of human endeavors and designs have an optimal aspiration in mind—minimize risk, maximize reward, reduce energy consumption, train a neural network to minimize model loss, *et cetera*.
- Nature’s phenomena themselves seem to be guided by optimization—the principle of least action, Hamiltonian mechanics, the Rayleigh-Ritz variational principle (from which the Schrödinger equation may be derived), *et cetera*.¹

But the benefits of an optimization mindset are also more subtle. For example, optimization in the space of semidefinite matrices has led to important breakthroughs in classical combinatorial optimization problems in graph theory, such as the MAX-CUT problem [GW95]. Optimization theory can also be used to show the *fundamental theorem of algebra*, and amusingly Güler presents that on page 34 (!) of his “Foundations of Optimization” book [Gül10], adapting some ideas by Terkelsen, F. [Ter76] and Fefferman, C. [Fef67].

The literature on optimization dates back to (at least) the ancient Greeks, with their work on the *isoperimetric problem*. The problem, which asks to construct the shape that maximizes the area given a fixed perimeter, is at the center of the legend of how Carthage—one of the most powerful empires of the classical world—was founded.² It was solved (to the standards of rigor of those times, at least) by Zenodorus.³

Fast forward to today, mathematical optimization is a vast field, subdivided into a myriad of subfields depending on what types of optimization problems are being solved; here are just a few:

- Linear optimization refers to problems that seek to minimize or maximize linear objectives subject to linear constraints imposed on continuous optimization variables.
- Integer linear optimization is akin linear optimization, but variables can be further restricted to only taking integer values.

*These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

¹See https://en.wikipedia.org/wiki/Variational_principle for a longer list.

²Dido was the legendary founder—and first queen—of Carthage, a Phoenician city-state that would then become of the most powerful empires in the world, and will be destroyed by the Roman empire. As recounted by Blåsjö, V. [Blå05], the legend says that “Dido, daughter of the king of Tyre, fled her home after her brother had killed her husband”. She later “ended up on the north coast of Africa, where she bargained to buy as much land as she could enclose with an oxhide. So she cut the hide into thin strips, and then she faced, and presumably solved, the isoperimetric problem.”

³Approx 150BC, [https://en.wikipedia.org/wiki/Zenodorus_\(mathematician\)](https://en.wikipedia.org/wiki/Zenodorus_(mathematician))

- Combinatorial optimization more generally studies optimization problems over finite sets.
- Convex optimization studies optimization of convex objectives on convex sets.
- Calculus of variations studies optimization problems in function spaces (that is, where the optimal object that is sought is a function).

1.1 Focus and objectives of this course

This is a graduate-level course on *nonlinear* optimization. The main focus of this course is *continuous optimization in finite dimensional spaces*. As such, nonlinear optimization encompasses a large variety of topics and algorithms, with applications that range from semidefinite relaxations of combinatorial problems, to regression problems, to training of deep neural networks. I foresee that this course attracts students from (at least) two very different categories:

- Some of you are interested in *nonlinear optimization theory*. Typically, you're facing an optimization problem that you hope to solve in closed form, so that you can plug in or reason about the solution as part of a larger mathematical endeavor.
- Some of you are interested in *nonlinear optimization algorithms*. Typically, you are facing an optimization problem for which you simply want a numerical solution.

In this course, the TAs and I have worked hard to cater to *both* of these categories of students.

1.2 Schedule and administrative details

Please see the syllabus available on Canvas for administrative details, contact information, homework and exam dates, and a tentative schedule of classes.

1.3 Prerequisites

This class is pretty light on prerequisites. Fundamentally, the only prerequisites are as follows:

- Linear algebra (18.06)
- Real analysis (18.100A/B/Q)

Beyond some basic mathematical concepts (continuity, vector spaces, some basic point-set topology), we will try to keep this course as self-contained as possible. Always feel free to stop and ask me or the TAs during class if there is something you are not familiar with!

A few homework problems will require some (simple!) coding. We recommend Python but you can ask to switch to a different language of your choice—but please, ask the TAs before doing so. If you have absolutely zero exposure to programming, that should not be a big issue, but please make sure to talk to the TAs about it this week.

2 Optimization problems and some preliminary considerations

2.1 General form of a nonlinear optimization problem

In general, an optimization problem has the form

$$\begin{aligned} \min_x & f(x) \\ \text{s.t. } & x \in \Omega \end{aligned}$$

In terms of nomenclature:

- The function we are trying to minimize (or maximize), in this case $f(x)$, is called the *objective function*.
- The entries of the vector x , are called *optimization variables*.

- The set Ω is called the *feasible set* or *constraint set*.

In particular, in this class we are interested in optimization problems that satisfy certain assumptions:

- First of all, we assume that Ω is a subset of some embedding *finite* Euclidean space E . Without loss of generality, $E = \mathbb{R}^n$.
- The objective function $f(x)$ is a *continuous* real-valued function. Very often, we also assume it is differentiable.
- A category of problems we are paying attention to are those with *functional constraints*, that is those in which the feasible set is defined as

$$\Omega := \left\{ x \in E \mid \begin{array}{l} g_i(x) \leq 0, \quad i = 1, \dots, r \\ h_j(x) = 0, \quad j = 1, \dots, m. \end{array} \right\}. \quad (1)$$

When faced with an optimization problem, two questions are immediate:

- Does a solution *exist*?
- If yes, can we *compute* it?

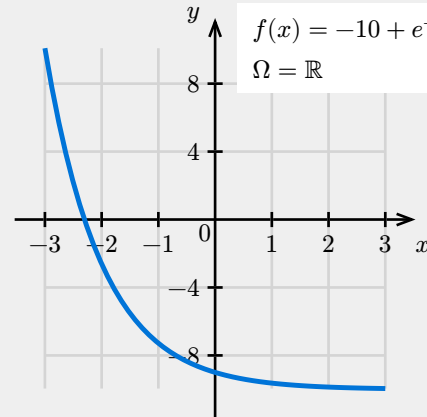
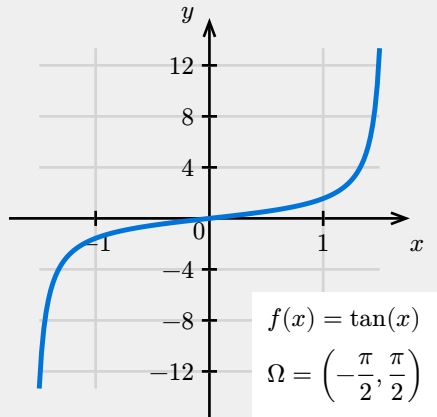
We will look into some preliminary observations regarding these two questions in Sections 2.2 and 2.3, respectively.

2.2 Existence of solutions: Weierstrass theorem

There are two reasons why an optimization problem might not attain a minimum:

- Ω is empty, that is, no feasible points exist at all; or
- for any candidate minimizer, there always exists another point that achieves a lower value, as shown in the following two examples.

Example 2.1. Consider minimization of the the following two functions, both over their domain:



In the plot on the left, the objective function is *unbounded*: values can be made arbitrarily small. No minimizer exists.

In the plot on the right, the objective has a lower bound, but for any value of x , all the values $x' > x$ achieve a *strictly smaller value* than $f(x)$. Again, no minimizer exists.

So, assuming that the objective f is continuous, when can we be certain that an optimizer exists? If we inspect Example 2.1, we notice that both functions are continuous. In the first example, the domain is open (the extrema $\pm\frac{\pi}{2}$ are not in the domain) and bounded. In the second example, the

domain is unbounded and closed. This shows that, if there is any hope of establishing a general result on existence of minimizers, such a result must require *both closedness and boundedness*. We are in luck:

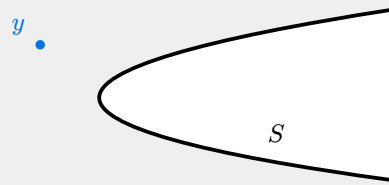
Theorem 2.1 (Weierstrass theorem). Let $f : \Omega \rightarrow \mathbb{R}$ be a *continuous* function defined on a non-empty and *compact* (i.e., closed and bounded) set Ω . Then, there exists a minimizer $x^* \in \Omega$ of f on Ω , that is,

$$f(x^*) \leq f(x) \quad \text{for all } x \in \Omega.$$

A note on boundedness. Weierstrass theorem is a pretty universal tool that you can keep in your toolbox for when you need to argue that an optimization problem has a solution. However, many optimization problems do not start off with a bounded feasible set Ω . Consider the following example:

Example 2.2. Let $y \in \mathbb{R}^n$ be a point, and $S \subseteq \mathbb{R}^n$ be a *closed* but not necessarily bounded set. Prove that there always exists a projection point

$$\begin{aligned} \min_x \quad & \|x - y\| \\ \text{s.t.} \quad & x \in S. \end{aligned}$$

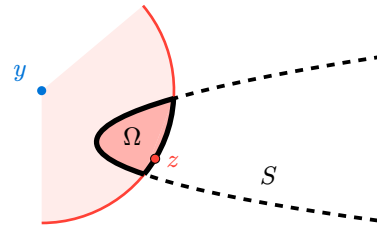


Solution. The idea is to show that we can safely restrict our attention to a compact subset of the feasible set of the problem—in this case, S . By “safely”, we mean that the minimizer does not change before and after making the modification.

In this case, this is simple: pick an arbitrary point $z \in S$. Then, we know that for sure, if a minimizer exists, it must live in the intersection

$$\Omega := S \cap \overline{\mathbb{B}}_{\|z-y\|}(y)$$

between S and the closed ball centered in y of radius $\|y - z\|$ (see the figure on the side). Now, Ω is nonempty (it contains z), closed (since both S and the ball are closed and intersection preserves closedness) and bounded (since the ball has finite radius). So, since the objective function is continuous, we can invoke Weierstrass theorem and conclude the proof. \square



More generally, we have the following:

Theorem 2.2 (Weierstrass theorem for compact sublevel sets). Let f be a continuous function defined on a set S . If f has a nonempty and compact *sublevel set*, that is, there exists $\alpha \in \mathbb{R}$ such that

$$\{x \in S : f(x) \leq \alpha\}$$

is nonempty and bounded,⁴ then f has a minimizing point in S .

⁴Closedness of the sublevel set is guaranteed by the continuity of f .

2.3 Nonlinear optimization is computationally hard

We now turn our attention to the second question: *what is reasonable to expect in terms of computational efficiency in finding the solution?*

Let me start with the good news: there are entire subclasses of nonlinear optimization problems that are computationally tractable (most notably, *convex* optimization problems). As we will see, even beyond these classes, usually it's possible to at least guarantee some form of convergence to some weaker, "local", notion of optimality.

However, in general, *finding minimizers of nonconvex problems is computationally intractable*. This means that we believe that there does not exist any algorithm that is able to solve *all* nonconvex optimization problems. This is because we believe that *linear integer programming* is already intractable, and nonconvex optimization captures known intractable problems such as MAX-CUT,⁵ already under benign assumptions such as quadratic functional constraints and objective.

Example 2.3 (MAX-CUT as a nonlinear optimization problem). MAX-CUT is one of the most fundamental problems in theoretical computer science and discrete optimization. The definition is very simple:

- Let $G = (V, E)$ be a graph, where $V := \{1, 2, \dots, n\}$ is the set of vertices.
- We want to partition V into two sets S and $V \setminus S$, such that *the number of edges going between S and $V \setminus S$ is maximized*.

Can you see how this optimization problem might be formulated as a nonlinear optimization problem with integer variables? Can you remove the constraint that the variables are integer, and convert it into a continuous problem?

Solution. As a general rule of thumb, the first we want to do when we go from a problem into its mathematical programming formulation, is to *pick variables*. In this case, it is intuitive that we need variables that encode whether each node belongs to S or $V \setminus S$. While several choices make perfect sense, let's go with the following. Let's assign to each node $v \in V$ a variable $x_v \in \{-1, +1\}$, with the meaning that:

- if $x_v = +1$, then $v \in S$;
- if $x_v = -1$, then $v \in V \setminus S$.

It is immediate to see that an edge $(u, v) \in E$ is a cut edge (that is, connecting one node in S to one node in $V \setminus S$) if and only if $x_u \cdot x_v = -1$. Else, $x_u \cdot x_v = +1$. Hence, we can count exactly all edges in the cut by considering the objective function

$$h(x) = \sum_{(u,v) \in E} \frac{1 - x_u \cdot x_v}{2}.$$

This leads to the *discrete* optimization problem

$$\begin{aligned} \max_x \quad & h(x) \\ \text{s.t.} \quad & x_v \in \{-1, +1\} \quad \forall v \in V. \end{aligned}$$

This problem can be readily written as a continuous, nonlinear optimization problem by noting that

$$x_v \in \{-1, +1\} \iff x_v^2 = 1.$$

⁵NP-completeness of MAX-CUT was shown by Garey, M. R., Johnson, D. S., & Stockmeyer, L. [GJS76].

This leads to the nonlinear optimization problem

$$\begin{aligned} \max_x \quad & \sum_{(u,v) \in E} \frac{1 - x_u \cdot x_v}{2} \\ \text{s.t.} \quad & x_v^2 = 1 \quad \forall v \in V, \end{aligned} \tag{2}$$

which is a quadratically-constrained quadratic optimization problem. \square

Despite the overall generally negative message, I want to end with a message of encouragement. Sure, we cannot hope to construct a machine that can solve all nonlinear optimization problems. But nonlinear optimization theory is still incredibly powerful, both because *many useful/powerful optimization problems can be solved efficiently*, and because *even for the hard problems, nonlinear optimization might have some tricks up its sleeve*. As a case in point, it turns out that the optimization problem in (2), which is hard, can be *relaxed* to a pretty non-obvious optimization problem that *can* be efficiently solved.

For those interested, this is obtained by introducing the relaxation

$$\begin{aligned} \max_y \quad & \sum_{(u,v) \in E} \frac{1 - y_{uv}}{2} \\ \text{s.t.} \quad & y_{vv} = 1 \quad \forall v \in V \\ & [y_{uv}] \succeq 0, \end{aligned}$$

where $[y_{uv}]$ denotes the matrix where the entry in row $u \in V$ and column $v \in V$ is y_{uv} and $\succeq 0$ means that the matrix just defined must be positive semidefinite. The important fact here is that this relaxation *can* be efficiently solved (we will see how later on in this course). With some clever rounding scheme, one can then recover a solution to the original MAX-CUT instance that is guaranteed to be 0.878-approximate. This algorithm, due to [Goemans, M. X., & Williamson, D. P. \[GW95\]](#) makes deep use of nonlinear optimization.

Bibliography

- [GW95] M. X. Goemans and D. P. Williamson, “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,” *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995, doi: [10.1145/227683.227684](https://doi.org/10.1145/227683.227684).
- [Gül10] O. Güler, *Foundations of Optimization*. New York, NY, USA: Springer, 2010. [Online]. Available: <https://link.springer.com/book/10.1007/978-0-387-68407-9>
- [Ter76] F. Terkelsen, “The Fundamental Theorem of Algebra,” *Am. Math. Mon.*, Oct. 1976, [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/00029890.1976.11994202>
- [Fef67] C. Fefferman, “An Easy Proof of the Fundamental Theorem of Algebra,” *The American Mathematical Monthly*, vol. 74, no. 7, pp. 854–855, 1967, [Online]. Available: <http://www.jstor.org/stable/2315823>
- [Blå05] V. Blåsjö, “The Isoperimetric Problem,” *Am. Math. Mon.*, Jun. 2005, [Online]. Available: <https://www.tandfonline.com/doi/pdf/10.1080/00029890.2005.11920227>
- [GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer, “Some simplified NP-complete graph problems,” *Theoret. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, Feb. 1976, doi: [10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).