ORF522 – Linear and Nonlinear Optimization

1. Introduction
What is this course about?

The mathematics behind making optimal decisions
What is this course about?

The mathematics behind making optimal decisions
What is this course about?

The mathematics behind making optimal decisions

Variables

Objective
What is this course about?

The mathematics behind making optimal decisions

- Variables
- Objective
- Constraints
What is this course about?

The mathematics behind making optimal decisions
Finance

Variables
Amounts invested in each asset

Constraints
Budget, investment per asset, minimum return, etc.

Objective
Maximize profit, minus risk
Optimal control

Variables
Inputs: thrust, flaps, etc.

Constraints
System limitations, obstacles, etc.

Objective
Minimize distance to target and fuel consumption
Machine learning

**Variables**
Model parameters

**Constraints**
Prior information, parameter limits

**Objective**
Minimize prediction error, plus regularization
Mathematical optimization

minimize \( f(x) \)
subject to \( g_i(x) \leq 0, \quad i = 1, \ldots, m \)

\[ x = (x_1, \ldots, x_n) \quad \text{Variables} \]
\[ f : \mathbb{R}^n \rightarrow \mathbb{R} \quad \text{Objective function} \]
\[ g_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad \text{Constraint functions} \]

\( x^* \quad \text{Solution/Optimal point} \)
\( f(x^*) \quad \text{Optimal value} \)
Most optimization problems cannot be solved
Solving optimization problems

General case → Very hard!

Compromises

• Long computation times
• Not finding the solution (in practice it may not matter)
Solving optimization problems

General case → Very hard!

Compromises

• Long computation times
• Not finding the solution (in practice it may not matter)

Exceptions

• Linear optimization
• Convex optimization

Can be solved very efficiently and reliably
Meet your teaching staff

Bartolomeo Stellato
I am a Professor at ORFE. I obtained my PhD from Oxford and I was a postdoc at MIT.

email: bstellato@princeton.edu
office hours: Thu 2pm—4pm EST, Sherred 323
website: stellato.io

Irina Wang
PhD student at ORFE.

e-mail: iywang@princeton.edu
office hours: Mon 1:00pm—3:00pm EST, at Sherrerd 003
Meet your classmates!

Name?  Year?

[QR Code]
Meet your classmates!

What is your department?

https://www.menti.com/5jp334nxuj
Meet your classmates!

Name?       Year?

What is your department?
https://www.menti.com/5jp334nxuj

What do you want to use optimization for?
Today’s agenda

• Optimization problems
• History of optimization
• Course contents and information
• A glance into modern optimization
Linear optimization

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

No analytical formula (99% of the time there will be none in this course!)

Efficient algorithms and software we can solve problems with several thousands of variables and constraints

Extensive theory (duality, degeneracy, sensitivity)
Linear optimization
Example: resource allocation

maximize \[ \sum_{i=1}^{n} c_{i} x_{i} \]
subject to \[ \sum_{i=1}^{n} a_{ji} x_{i} \leq b_{j}, \quad j = 1, \ldots, m \]
\[ x_{i} \geq 0, \quad i = 1, \ldots, n \]

- \( c_{i} \): profit per unit of product \( i \) shipped
- \( b_{j} \): units of raw material \( j \) on hand
- \( a_{ji} \): units of raw material \( j \) required to produce one unit of product \( i \)
Nonlinear optimization

minimize $f(x)$
subject to $g_i(x) \leq 0, \quad i = 1, \ldots, m$

Hard to solve in general

• multiple local minima
• discrete variables $x \in \mathbb{Z}^n$
• hard to certify optimality
Convex optimization

Convex functions

minimize $f(x)$
subject to $g_i(x) \leq 0, \ i = 1, \ldots, m$

All local minima are global!

Efficient algorithms and software

Extensive theory (convex analysis and conic optimization) [ORF523]

Used to solve non convex problems
# Prehistory of optimization

## Calculus of variations

<table>
<thead>
<tr>
<th>Fermat/Newton</th>
<th>Euler</th>
<th>Lagrange</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize ( f(x), \ x \in \mathbb{R} )</td>
<td>minimize ( f(x), \ x \in \mathbb{R}^n )</td>
<td>minimize ( f(x) ) subject to ( g(x) = 0 )</td>
</tr>
<tr>
<td>( \frac{df(x)}{dx} = 0 )</td>
<td>( \nabla f(x) = 0 )</td>
<td></td>
</tr>
<tr>
<td>1670</td>
<td>1755</td>
<td>1797 Time</td>
</tr>
</tbody>
</table>

1670

1755

1797

Time
History of optimization

Origin of linear optimization (Kantorovich, Koopmans, von Neumann) 1930s

Simplex algorithm (Dantzig) 1947

Interior-point methods (Karmarkar) 1984

Large-scale optimization 2000s
History of optimization

### Algorithms
- **Origin of linear optimization**: Kantorovich, Koopmans, von Neumann (1930s)
- **Simplex algorithm**: Dantzig (1947)
- **Interior-point methods**: Karmarkar (1984)
- **Large-scale optimization**: 2000s

### Applications
- **Operations Research, Economics**: 1990s
- **Engineering, Statistics**: 2000s
- **Machine learning, Image processing, Communication systems, Embedded intelligent systems**: 2000s
History of optimization

Origin of linear optimization
(Kantorovich, Koopmans, von Neumann)

Simplex algorithm
(Dantzig)

Interior-point methods
(Karmarkar)

Large-scale optimization

1930s 1947 1984 2000s

Algorithms

Operations Research
Economics

Engineering
Statistics

Applications

Machine learning
Image processing
Communication systems
Embedded intelligent systems

1990s 2000s
Technological innovations

Lots of data

easy storage
and
transmission
Technological innovations

Lots of data

Massive computations

easy storage and transmission

computers are super fast
Technological innovations

Lots of data
- easy storage and transmission

Massive computations
- computers are super fast

High-level programming languages
- easy to do complex stuff
What is happening today?

Huge scale optimization
- Massive data
- Massive computations

Real-time optimization
- Fast real-time requirements
- Low-cost computing platforms
What is happening today?

Huge scale optimization
Massive data
Massive computations

Real-time optimization
Fast real-time requirements
Low-cost computing platforms

Renewed interest in old methods (70s)
• Subgradient methods
• Proximal algorithms
What is happening today?

Huge scale optimization
- Massive data
- Massive computations

Real-time optimization
- Fast real-time requirements
- Low-cost computing platforms

Renewed interest in old methods (70s)
- Subgradient methods
- Proximal algorithms

Cheap iterations
Simple implementation
Contents of this course

Linear optimization
- Modelling and applications
- Geometry
- Duality
- Degeneracy
- The simplex method
- Sensitivity analysis
- Interior point methods

Nonlinear optimization
- Modelling and applications
- Optimality conditions
- First-order methods
- Operator-splitting algorithms
- Acceleration schemes

Extensions
- Sequential convex programming
- Branch and bound algorithms
- Real-time optimization
Course information
Grading

• 25% Homeworks
  5 bi-weekly homeworks with coding component. Collaborations are encouraged!

• 25% Midterm
  90 minutes written exam. No collaborations.

• 40% Final
  Take-home assignment with coding component. No collaborations.

• 10% Participation
  One question or note on Ed after each lecture.
Course information
10% Participation notes/questions

What?

• Briefly summarize what you learned in the last lecture

• Highlight the concepts that were most confusing/you would like to review.

• Can be anonymous (to your classmates, not to the instructor) or public, as you choose.

Why?

• We will use your ideas to clarify previous lectures, and to improve the course in future iterations.

• You can ask questions you don’t feel comfortable asking in class.

• You can use these to gather your thoughts on the previous lecture and solidify your understanding.
Course information

Course website
https://stellato.io/teaching/orf522

Prerequisites

- Good knowledge of linear algebra and calculus.
  For a refresher, read Appendices A & C of [CO] Boyd, Vandenberghe: Convex Optimization (available online).

- Familiarity with Python.
Course information

Materials

Linear optimization

Nonlinear optimization
- [NO] J. Nocedal, S. J. Wright: *Numerical Optimization* (available on SpringerLink)
- [FMO] A. Beck: *First-order methods in optimization* (available on SIAM)
- [ILCO] Y. Nesterov: *Introductory Lectures to Convex Optimization* (available on SpringerLink)
- [e364b] S. Boyd: *Convex Optimization II Lecture Notes* (available online)

Operator splitting algorithms
- [PA] N. Parikh, S. Boyd: *Proximal Algorithms* (available for free)
- [PMO] E. K. Ryu, S. Boyd: *A primer on monotone operators* (available for free)
- [ADMM] S. Boyd, N. Parikh, B. Peleato, J. Eckstein: *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* (available for free)
Numerical computations
Numerical computations on *numpy* and *scipy*.

CVXPY

minimize \( c^T x \)
subject to \( Ax \leq b \)

```python
x = cp.Variable(n)
prob = cp.Problem(
    cp.Minimize(c.T@x),
    [A @ x <= b])
prob.solve()
print("The optimal value is", prob.value)
print("The solution x is", x.value)
```
Learning goals

- **Model** your favorite decision-making problems as mathematical optimization problems.

- **Apply** the most appropriate optimization tools when faced with a concrete problem.

- **Implement** optimization algorithms and prove their convergence.
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\)  \rightarrow \textbf{Goal:} Design predictor \(\hat{y}^i = g_\theta(x^i)\)
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\)  

Goal: Design predictor \(\hat{y}^i = g_\theta(x^i)\)

Optimization problem

\[
\text{minimize} \quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)
\]
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\) \(\rightarrow\) \textbf{Goal:} Design predictor \(\hat{y}^i = g_\theta(x^i)\)

Optimization problem

\[\text{minimize} \quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)\]
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\)  \(\rightarrow\) **Goal:** Design predictor \(\hat{y}^i = g_\theta(x^i)\)

**Optimization problem**

\[
\text{minimize} \quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)
\]
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\) \quad \longrightarrow \quad \textbf{Goal:} Design predictor \(\hat{y}^i = g_\theta(x^i)\)

\[
\begin{align*}
\text{Optimization problem} & \\
\text{Loss} & + \text{Regularizer} \\
\text{minimize} & \quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)
\end{align*}
\]

Many examples

- Support vector machines
- Regularized regression
- Neural networks
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\)  
\(\rightarrow\)  
**Goal:** Design predictor \(\hat{y}^i = g_\theta(x^i)\)

Optimization problem

\[
\begin{align*}
\text{minimize} & \quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta) \\
\end{align*}
\]

**Many examples**
- Support vector machines
- Regularized regression
- Neural networks

**Large-scale computing**
- Parallel
- Distributed
Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints \((x^i, y^i)\)  \(\rightarrow\) \textbf{Goal:} Design predictor \(\hat{y}^i = g_\theta(x^i)\)

Optimization problem

\[
\begin{align*}
\text{minimize} & & \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta) \\
\end{align*}
\]

Many examples

- Support vector machines
- Regularized regression
- Neural networks

Large-scale computing

- Parallel
- Distributed

How large are the largest problems we can solve? (how many variables?)
Glance into modern optimization

Real-time optimization

Dynamical system: \( x_{t+1} = Ax_t + Bu_t \)

\( x_t \in \mathbb{R}^n : \) state

\( u_t \in \mathbb{R}^m : \) input

Goal: track trajectory

\[
\min \sum_{t=0}^{T} \| x_t - x_t^{\text{des}} \|
\]

Constraints: inputs \( \| u \| \leq U \), states \( a \leq x_t \leq b \)
Glance into modern optimization

Real-time optimization

Dynamical system: \[ x_{t+1} = Ax_t + Bu_t \]
- \( x_t \in \mathbb{R}^n \): state
- \( u_t \in \mathbb{R}^m \): input

**Goal:** track trajectory \[ \text{minimize } \sum_{t=0}^{T} \| x_t - x_{t}^{\text{des}} \| \]

**Constraints:** inputs \[ \| u \| \leq U \], states \[ a \leq x_t \leq b \]

Solve and repeat.....

How fast can we solve these problems?
Glance into modern optimization

Real-time optimization

Dynamical system: \( x_{t+1} = Ax_t + Bu_t \)

\( x_t \in \mathbb{R}^n : \) state

\( u_t \in \mathbb{R}^m : \) input

**Goal:** track trajectory minimize \( \sum_{t=0}^{T} ||x_t - x_t^{\text{des}}|| \)

**Constraints:** inputs \( ||u|| \leq U \), states \( a \leq x_t \leq b \)

Solve and repeat…..

How fast can we solve these problems?

1-norm \( \rightarrow \) ???
Glance into modern optimization

Real-time optimization

Dynamical system: \( x_{t+1} = Ax_t + Bu_t \)

- \( x_t \in \mathbb{R}^n \): state
- \( u_t \in \mathbb{R}^m \): input

**Goal**: track trajectory minimize \( \sum_{t=0}^{T} \| x_t - x_t^{\text{des}} \| \)

**Constraints**: inputs \( \| u \| \leq U \), states \( a \leq x_t \leq b \)

Solve and repeat.....

How fast can we solve these problems?
Next lecture
Linear optimization

• Definitions
• Modelling
• Formulations
• Examples