CONVEX OPTIMIZATION

- Venue and time
- Topics and teachers
- Prerequisites
    - Calculus
    - Linear Algebra
    - Probability theory
    - Basic Matlab
- Materials
    - Stephen Boyd https://web.stanford.edu/class/ee364a/
    - Class Notes : Canvas : \Files\ notes
- Homeworks → 30%
- Final Exam / Course Project → 70%

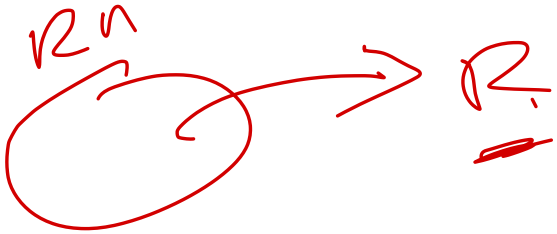
IKT720-G 20H: Optimization (Wednesday & Friday: 11.15-13.00)

Dates	Lecture	Lecturer	HomeWorks
16 Sept	Introduction, Convex sets (1/2)	Joshin	
18 Sept	Convex sets (2/2), Convex functions (1/2)	Joshin	
23 Sept	Convex functions (2/2)	Joshin	HW1
25 Sept	Convex Problems (1/2)	Joshin	
07 Oct	Convex Problems (2/2)	Joshin	
09 Oct	HW1 answers	Emilio, Juan	HW2
14 Oct	CVXPY - introduction	Emilio, Ravi	
16 Oct	Duality (1/3)	Emilio	
21 Oct	Duality (2/3)	Emilio	
23 Oct	Duality (3/3)	Emilio	HW3
28 Oct	Applications 1 (Signal processing, fitting, filter design)	Siddharth	
30 Oct	Applications 2 (Statistical estimation)	Siddharth	
04 Nov	Application 3 (Classification, SVM, ML)	Siddharth	Project Proposal
06 Nov	CVXPY - applications	Emilio, Ravi	
11 Nov	Algebra Background, HW2 answers	Luismi, Emilio	
13 Nov	Iterative algorithms I	Luismi	
18 Nov	Iterative algorithms II	Luismi	
20 Nov	Iterative algorithms III	Luismi	HW4
25 Nov	HW3 answers	Emilio	
27 Nov	Project Delivery	All	
02 Dec	Final Exam	All	

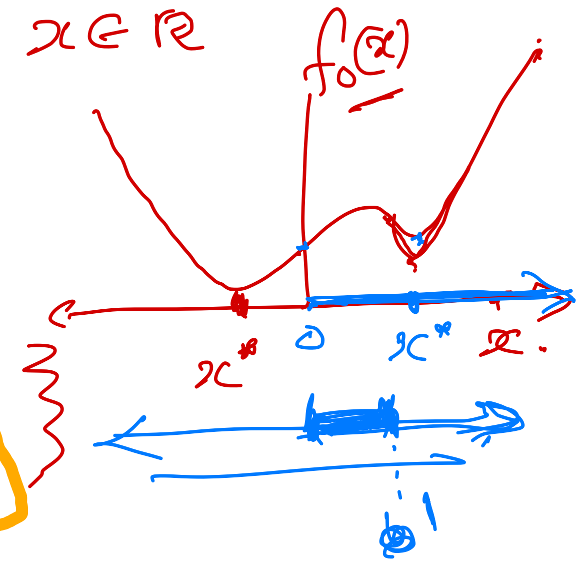
See Canvas : Files\ Course Info

1. Introduction

- mathematical optimization
- least-squares and linear programming
- convex optimization
- example
- course goals and topics
- ~~nonlinear optimization~~
- brief history of convex optimization



Mathematical optimization



(mathematical) optimization problem

$x_1+x_2+x_3$

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le b_i, \quad i = 1, \dots, m \end{array}$$

$x \ge 0$

$x \le 1$

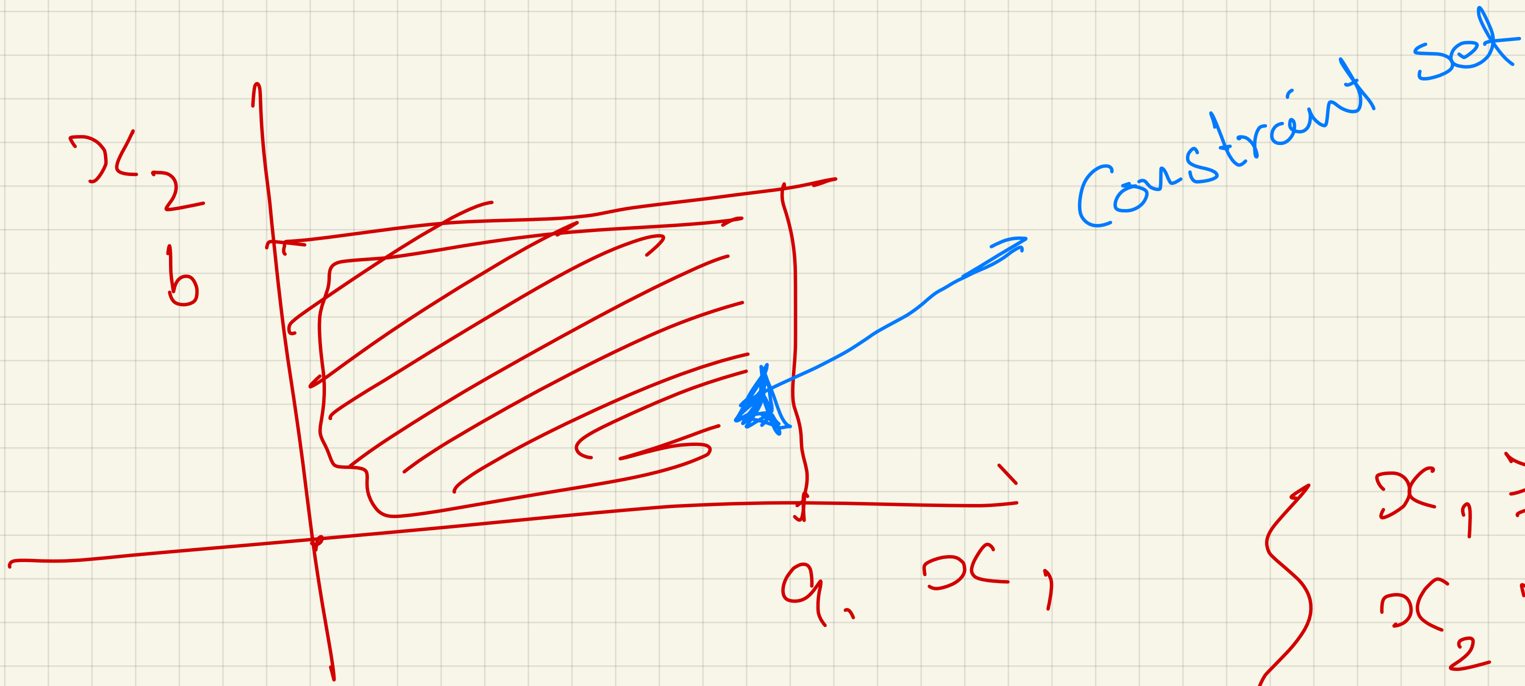
- $x = (x_1, \dots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \dots, m$: constraint functions

max ≡ min

solution or **optimal point** $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

objective fn but No constraints → Unconstrained optimizn problem

constraints but No objective function → feasibility problem



$$\begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \\ x_1 \leq a \\ x_2 \leq b. \end{cases}$$

$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\quad c^T x = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Examples

$x_i \Rightarrow$ no of stocks

x_1, x_2, x_3

risk $= f(x_1, x_2, x_3) \rightarrow$ objective function

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

cost $c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$

$c^T x \leq 1000$ NOK $\rightarrow$ constraint

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error, plus regularization term

economics, physics

Why studying Optimization ?

- Use of Optimization ubiquitous: Communic., Signal Process., Machine Learning, ...

- Many practical problems can be cast or converted into optimization-like problems

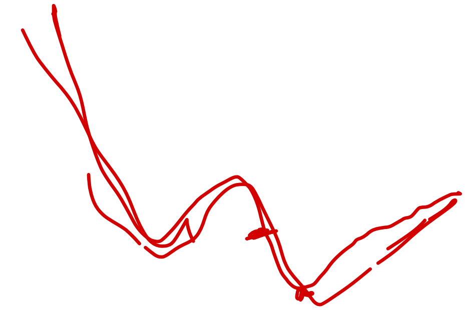
- Three different major waves in optimization theory and its applications:
  - Linear Programming (Linear Convex Optimization), late 1940s
  - General Convex Optimization, late 1980s
  - Non-convex Optimization, 1990s

- Each wave has been followed by a period of "appreciation-application cycle"
  - More people appreciate the use of a tool $\Longrightarrow$ more look for formulations in applic.
  - Then, more work on its theory, efficient algorithms and softwares
  - Then, more powerful the tools become $\Longrightarrow$ more people turn to appreciate its use
  - This very beneficial cycle has transformed indeed each of these waves.

- A lot of research going on in Convex and (specially) in Non-Convex Optimization (many research papers being published as we speak and several problems still open)

- Many applications benefit enormously from these waves (there is a huge array of many success stories in many different applications)

Solving optimization problems



general optimization problem

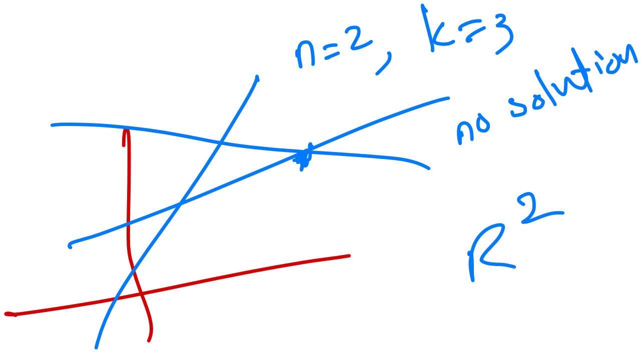
- very difficult to solve
- methods involve some compromise, $e.g.$, very long computation time, or not always finding the solution (which may not matter in practice)

exceptions: certain problem classes can be solved efficiently and reliably

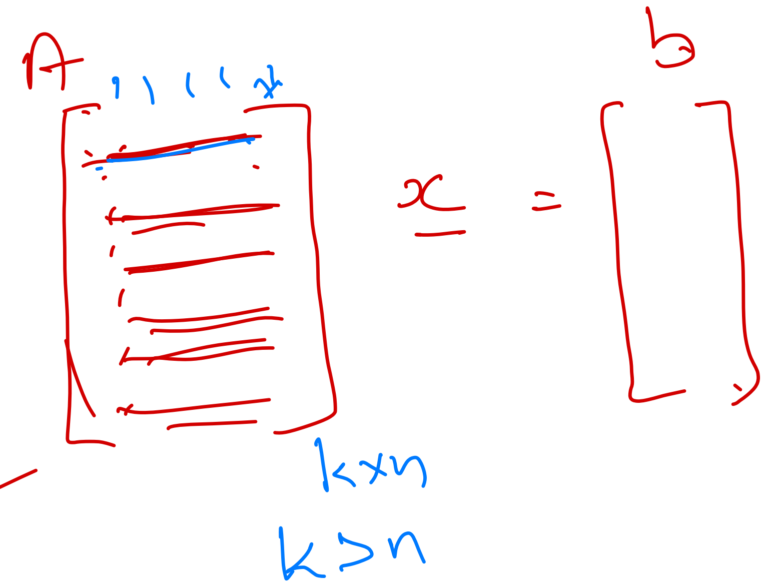
- least-squares problems
- linear programming problems
- convex optimization problems

CVX → convex

Note: We use 'cvx' to denote convex in the lectures. But it is an informal writing and should not be confused with the CVX matlab package that we discuss later in the course.



Least-squares



$$\text{minimize} \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $\boxed{x^\star = (A^TA)^{-1}A^Tb}$
- reliable and efficient algorithms and software
- computation time proportional to n^2k ($A \in \mathbf{R}^{k\times n}$); less if structured
- a mature technology

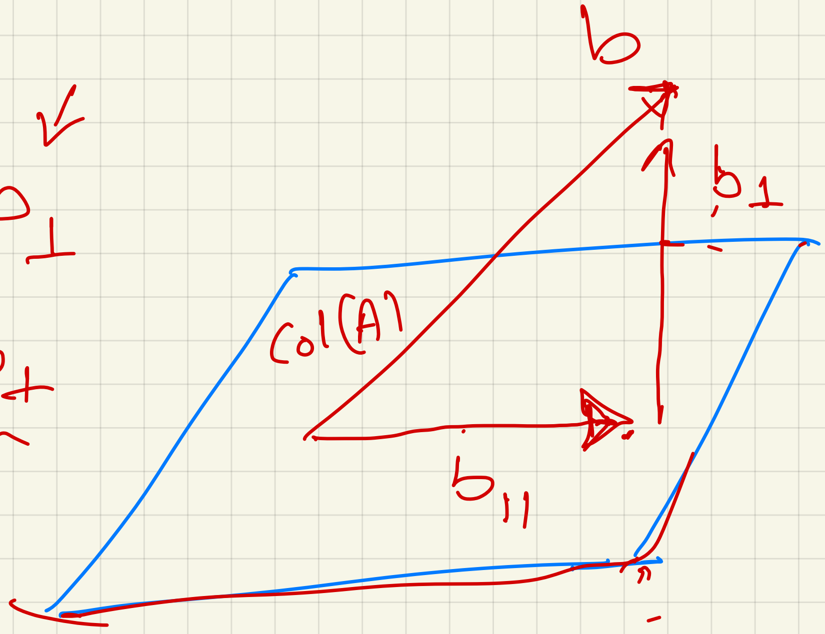
using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility ($e.g.$, including weights, adding regularization terms)

$$A\underline{x} = \underline{b}$$

$$b = b_{\parallel} + b_{\perp}$$

$$Ax = b_{\parallel} = b - b_{\perp}$$



$$\underline{A^T A x = A^T b} - \underbrace{A^T b_\perp}_{0}$$

$$\boxed{x = (A^T A)^{-1} A^T b}$$

$$\frac{d}{dx}\left(x^T A x\right)$$

Note: in the class I forgot to say that $\frac{d}{dx}\|Ax-b\|_2^2 = 0$. Of course you have to equate it to zero to find the minimum.

$\Downarrow$ derive this

$x \in \mathbb{R}^n$ $\quad x \in \mathbb{R}^{n \times 1}$

$$\|x\|_2^2 = \sqrt{x_1^2 + x_2^2 + x_3^2 \cdots}$$

$$\frac{d}{dx}\underline{\|Ax - b\|^2}$$

$$\frac{d}{dx}\underline{(Ax-b)^T (Ax-b)}$$

Linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

linear

linear

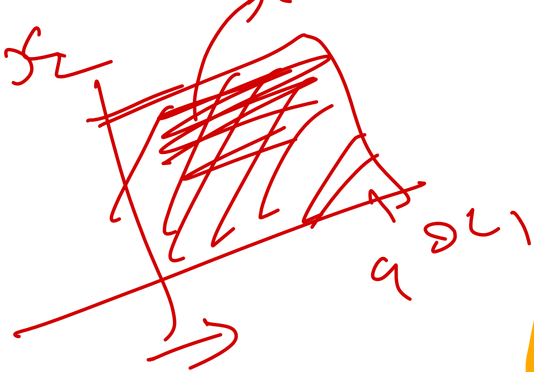
solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

constraint set



1 objective function

→ cvx

2 constraint set

→ cvx

Convex optimization problem

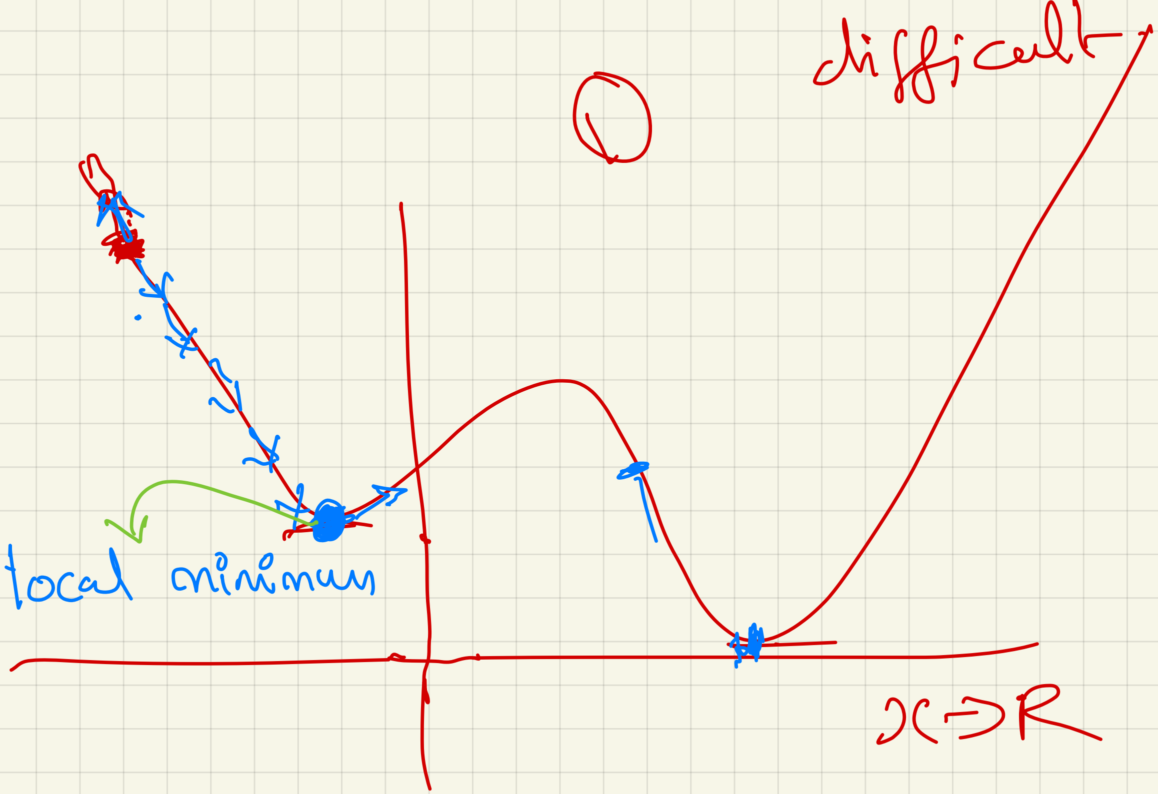
$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \le b_i, \quad i = 1, \ldots, m
\end{array}
$$

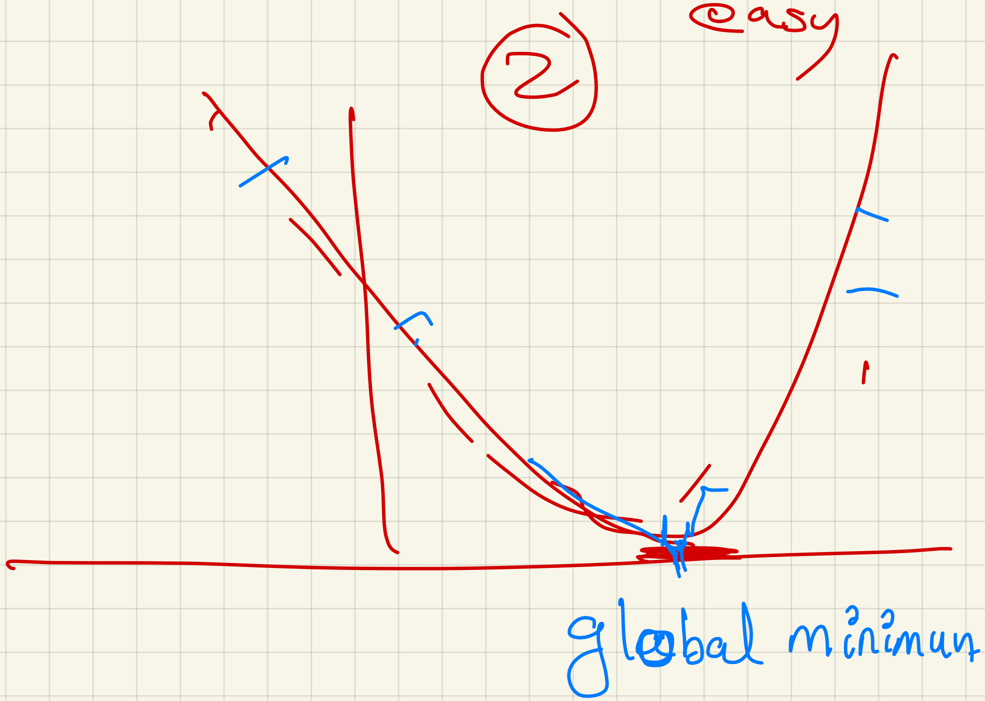
- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1$, $\alpha \ge 0$, $\beta \ge 0$

- includes least-squares problems and linear programs as special cases



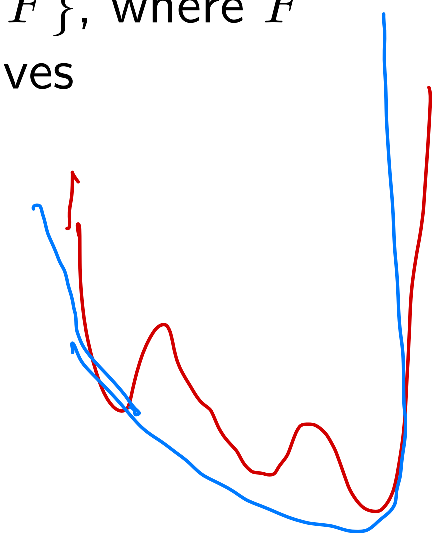


$$x^* = (A^T A)^{-1} A^T b$$

minimize $\|Ax-b\|_2^2$

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

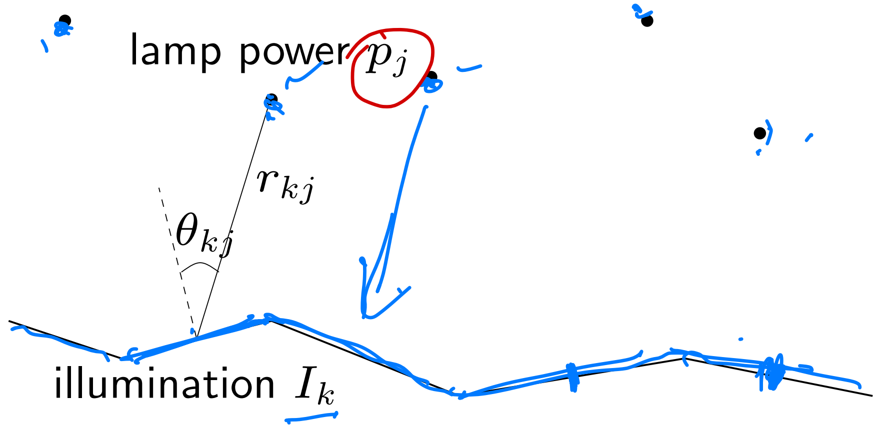


using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

Example

m lamps illuminating n (small, flat) patches



intensity I_k at patch k depends linearly on lamp powers p_j:

$$I_k = \sum_{j=1}^{m} a_{kj} p_j, \qquad a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}$$

$\log\left(\frac{I_k}{I_{des}}\right)$

problem: achieve desired illumination I_{des} with bounded lamp powers

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |\log I_k - \log I_{\text{des}}| \\ \text{subject to} & 0 \le p_j \le p_{\text{max}}, \quad j = 1, \ldots, m \end{array}$$

100

50

how to solve?

1. use uniform power: $p_j = p$, vary p

2. use least-squares:

$P_{max} = 100$

$P_j = 105 \to 100$

$P_j = -5 \to 0$

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2$$

   round p_j if $p_j > \overset{100}{p_{\text{max}}}$ or $p_j < 0$

$P_j < -5$

3. use weighted least-squares:

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2 + \sum_{j=1}^{m} w_j (p_j - p_{\text{max}}/2)^2$$

   iteratively adjust weights w_j until $0 \le p_j \le p_{\text{max}}$

4. use linear programming:

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |I_k - I_{\text{des}}| \\ \text{subject to} & 0 \le p_j \le p_{\text{max}}, \quad j = 1, \ldots, m \end{array}$$

   which can be solved via linear programming

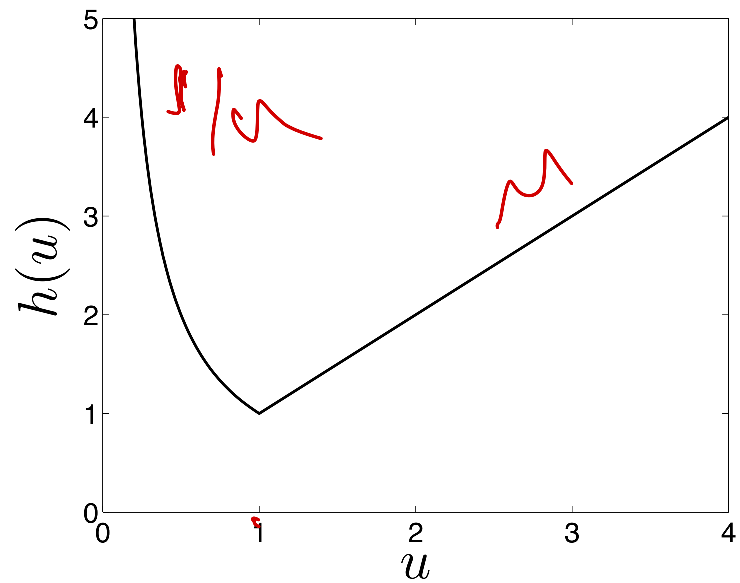
of course these are approximate (suboptimal) 'solutions'

5. use convex optimization: problem is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} h(I_k/I_{\mathrm{des}}) \\ \text{subject to} & 0 \le p_j \le p_{\max}, \quad j = 1,\ldots,m \end{array}$$

log

with $h(u) = \max\{u, 1/u\}$



f_0 is convex because maximum of convex functions is convex

exact solution obtained with effort $\approx$ modest factor $\times$ least-squares effort

$$\left| \log \frac{I_k}{I_{des}} \right| = \max \left\{ \log \frac{I_k}{I_{des}}, -\log \frac{I_k}{I_{des}} \right\}$$

$$-\log \frac{I_k}{I_{des}} = \log \frac{I_{des}}{I_k}$$

$$|y| = \max\{y, -y\}$$

$$\min \log x$$

$$\Rightarrow \min x$$

additional constraints: does adding 1 or 2 below complicate the problem?

1. no more than half of total power is in any 10 lamps
2. no more than half of the lamps are on ($p_j > 0$)

- answer: with (1), still easy to solve; with (2), extremely difficult
- moral: (untrained) intuition doesn't always work; without the proper background very easy problems can appear quite similar to very difficult problems

Brief history of convex optimization

theory (convex analysis): 1900–1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1970s: ellipsoid method and other subgradient methods
- 1980s & 90s: polynomial-time interior-point methods for convex optimization (Karmarkar 1984, Nesterov & Nemirovski 1994)
- since 2000s: many methods for large-scale convex optimization

applications

- before 1990: mostly in operations research, a few in engineering
- since 1990: many applications in engineering (control, signal processing, communications, circuit design, . . .)
- since 2000s: machine learning and statistics

What is this course all about ?

- **Goal:**
  Study of optimization tools that are useful in several areas:
  - **Acquire correctly the mentality and language of Optimization.**
    (surprisingly, this can be useful for other studies you may engage in later on)
  - **Formulate problems in these areas as optimization problems.**
  - **Understanding underlying theory, concepts and properties of optimization tools**
  - **Design, implement and simulate practical (centralized & distributed) algorithms to solve optimization problems.**
  - **Analyze the structures/decompositions of problems and solutions, as well as the relationship between different problems**
- **The course will cover both classic results as well as more recent results (over a wide range of problems)**
- **We will try to present the appropriate background material "just-in-time"**
- **Training also the ability to do original research & innovation in academia or industry**

What this course is NOT about

- Not a Math course on Convex Analysis
  (we will not do many rigorous proofs, and will skip also many proofs)

- Not an OR course on Non-linear Optimization
  (we will cover various additional topics + connection to applications)

- Not a EE course on Digital Communications or Signal Processing course
  (other Master courses cover these areas)

- Not an EE course on Networking
  (we will cover only selected topics, other Master courses cover Networking)

- Not a CS course on Algorithms and Complexity
  (we will not do an extensive computational complexity analysis)

- Not a CS or EE course on Machine Learning
  (we will cover only methods to formulate & solve some machine learning problems)