ORF522 – Linear and Nonlinear Optimization

2. Linear optimization
Today’s agenda
Readings: [Chapter 1, Bertsimas, Tsitsiklis]

• Linear optimization in inner-product and matrix notation
• Optimization terminology
• Standard form
• Piecewise-linear minimization
• Examples
Where does linear optimization appear?

Supply chain management
Assignment problems
Scheduling and routing problems
Finance
Optimal control problems
Network design and network operations
Many other domains…
Vector notations

By default, all vectors are column vectors and denoted by

\[ x = (x_1, \ldots, x_n) \]

The transpose of a vector is \( x^T \)

\[ a^T x \text{ is the inner product between } a \text{ and } x \]

\[ a^T x = a_1 x_1 + \cdots + a_n x_n = \sum_{i=1}^{n} a_i x_i \]
Linear optimization
Linear Programming (LP)

minimize
\[ \sum_{i=1}^{n} c_i x_i \]
subject to
\[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \]
\[ \sum_{j=1}^{n} d_{ij} x_j = f_i, \quad i = 1, \ldots, p \]

Objective function and constraints are \textit{linear in the decision variables}

Belongs to \textbf{continuous optimization}
Linear optimization

Inner product notation

minimize \[ \sum_{i=1}^{n} c_i x_i \]
subject to \[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \]
\[ \sum_{j=1}^{n} d_{ij} x_j = f_i, \quad i = 1, \ldots, p \]

minimize \[ c^T x \]
subject to \[ a_i^T x \leq b_i, \quad i = 1, \ldots, m \]
\[ d_i^T x = f_i, \quad i = 1, \ldots, p \]

\( c, a_i, d_i \) are \( n \)-vectors
\[ c = (c_1, \ldots, c_n) \]
\[ a_i = (a_{i1}, \ldots, a_{in}) \]
\[ d_i = (d_{i1}, \ldots, d_{in}) \]
Linear optimization
Matrix notation

minimize $\sum_{i=1}^{n} c_i x_i$
subject to
$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m$
$\sum_{j=1}^{n} d_{ij} x_j = f_i, \quad i = 1, \ldots, p$

$\rightarrow$

minimize $c^T x$
subject to
$A x \leq b$
$D x = f$

$A$ is $m \times n$-matrix with elements $a_{ij}$ and rows $a_i^T$
$D$ is $p \times n$-matrix with elements $d_{ij}$ and rows $d_i^T$
All (in)equalities are elementwise
Optimization terminology

minimize $c^T x$
subject to $Ax \leq b$
$Dx = f$

$x$ is feasible if it satisfies the constraints $Ax \leq b$ and $Dx = f$

The feasible set is the set of all feasible points

$x^*$ is optimal if it is feasible and $c^T x^* \leq c^T x$ for all feasible $x$

The optimal value is $p^* = c^T x^*$

Unbounded problem: $c^T x$ is unbounded below on the feasible set ($p^* = -\infty$)

Infeasible problem: feasible set is empty ($p^* = +\infty$)
Standard form

Definition

minimize \[ c^T x \]
subject to \[ Ax = b \]
\[ x \geq 0 \]

- Minimization
- Equality constraints
- Nonnegative variables

- Matrix notation for theory
- Standard form for algorithms
Standard form

Transformation tricks

Change objective
   If “maximize”, use $-c$ instead of $c$ and change to “minimize”.

Eliminate inequality constraints
   If $Ax \leq b$, define $s$ and write $Ax + s = b$, $s \geq 0$.
   If $Ax \geq b$, define $s$ and write $Ax - s = b$, $s \geq 0$. $s$ are the slack variables

Change variable signs
   If $x_i \leq 0$, define $y_i = -x_i$.

Eliminate “free” variables
   If $x_i$ unconstrained, define $x_i = x_i^+ - x_i^-$, with $x_i^+ \geq 0$ and $x_i^- \geq 0$. 
Standard form
Transformation example

minimize \[ 2x_1 + 4x_2 \]
subject to \[ x_1 + x_2 \geq 3 \]
\[ 3x_1 + 2x_2 = 14 \]
\[ x_1 \geq 0 \]

\[
\begin{align*}
\text{minimize} & \quad 2x_1 + 4x_2^+ - 4x_2^- \\
\text{subject to} & \quad x_1 + x_2^+ - x_2^- - x_3 = 3 \\
& \quad 3x_1 + 2x_2^+ - 2x_2^- = 14 \\
& \quad x_1, x_2^+, x_2^-, x_2^-, x_3 \geq 0.
\end{align*}
\]
Linear, affine and convex functions

Linear function: \( f(x) = a^T x \)
\[
f(\alpha x + \beta y) = \alpha f(x) + \beta f(y), \quad \forall x, y \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}
\]

Affine function: \( f(x) = a^T x + b \)
\[
f(\alpha x + (1 - \alpha)y) = \alpha f(x) + (1 - \alpha)f(y), \quad \forall x, y \in \mathbb{R}^n, \alpha \in \mathbb{R}
\]

Convex function:
\[
f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall x, y \in \mathbb{R}^n, \alpha \in [0, 1]
\]
Convex piecewise-linear functions

\[ f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i) \]
Convex piecewise-linear minimization

minimize \[ f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i) \]

Equivalent linear optimization

minimize \[ t \]
subject to \[ a_i^T x + b_i \leq t, \quad i = 1, \ldots, m \]

Matrix notation

minimize \[ \tilde{c}^T \tilde{x} \]
subject to \[ \tilde{A} \tilde{x} \leq \tilde{b} \]

\[ \tilde{x} = \begin{bmatrix} x \\ t \end{bmatrix}, \quad \tilde{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} a_1^T & -1 \\ \vdots & \vdots \\ a_m^T & -1 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} -b_1 \\ \vdots \\ -b_m \end{bmatrix} \]
Sum of piecewise-linear functions

minimize \( f(x) + g(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i) + \max_{i=1,\ldots,p} (c_i^T x + d_i) \)

Cost function is piecewise-linear

\[
f(x) + g(x) = \max_{i=1,\ldots,m} \left( (a_i + c_j)^T x + (b_i + d_j) \right)
\]

Equivalent linear optimization

\[
\begin{align*}
\text{minimize} & \quad t_1 + t_2 \\
\text{subject to} & \quad a_i^T x + b_i \leq t_1, \quad i = 1, \ldots, m \\
& \quad c_i^T x + d_i \leq t_2, \quad i = 1, \ldots, p
\end{align*}
\]

Matrix notation?
Examples
Cheapest cat food problem

- Choose quantities \( x_1, \ldots, x_n \) of \( n \) ingredients each with unit cost \( c_j \).
- Each ingredient \( j \) has nutritional content \( a_{ij} \) for nutrient \( i \).
- Require a minimum level \( b_i \) for each nutrient \( i \).

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1 \ldots m \\
& \quad x_j \geq 0, \quad j = 1 \ldots n
\end{align*}
\]

Would you give her the optimal food?
Data-fitting example

Fit a linear function \( f(z) = a + b z \) to \( m \) data points \((z_i, f_i)\):

Approximation problem \( A x \approx b \) where

\[
\begin{bmatrix}
1 & z_1 \\
\vdots & \vdots \\
1 & z_m
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
\approx
\begin{bmatrix}
f_1 \\
\vdots \\
f_m
\end{bmatrix}
\]
Data-fitting example

Fit a linear function \( f(z) = a + bz \) to \( m \) data points \((z_i, f_i)\):

Approximation problem \( Ax \approx b \) where

\[
\begin{bmatrix}
1 & z_1 \\
\vdots & \vdots \\
1 & z_m
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
\approx
\begin{bmatrix}
f_1 \\
\vdots \\
f_m
\end{bmatrix}
\]

Least squares way:

minimize \( \sum_{i=1}^{m} (Ax - b)_i^2 = \|Ax - b\|_2^2 \)

**Good news:** solution is in closed form \( x^* = (A^T A)^{-1} A^T b \)

**Bad news:** solution is very sensitive to outliers!
Data-fitting example

Fit a linear function \( f(z) = a + bz \) to \( m \) data points \((z_i, f_i)\):

Approximation problem \( Ax \approx b \) where

\[
\begin{bmatrix}
1 & z_1 \\
\vdots & \vdots \\
1 & z_m \\
\end{bmatrix}
\begin{bmatrix}
a \\
b \\
\end{bmatrix}
\approx
\begin{bmatrix}
f_1 \\
\vdots \\
f_m \\
\end{bmatrix}
\]

A different way:

minimize \( \sum_{i=1}^{m} |Ax - b|_i = \|Ax - b\|_1 \)

**Good news:** solution is much more robust to outliers.

**Bad news:** there is no closed form solution.
1-norm approximation

minimize $\|Ax - b\|_1$

The 1-norm of $m$-vector $y$ is

$$\|y\|_1 = \sum_{i=1}^{m} |y_i| = \sum_{i=1}^{m} \max\{y_i, -y_i\}$$

Equivalent problem

minimize $\sum_{i=1}^{m} u_i$

subject to $-u \leq Ax - b \leq u$

Matrix notation

minimize

$$\begin{bmatrix} 0^T \\ 1 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

subject to

$$\begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix}$$
Comparison with least-squares

Histogram of residuals $Ax - b$ with randomly generated $A \in \mathbb{R}^{200 \times 80}$

$x_{1s} = \arg\min ||Ax - b||$, $\quad x_{\ell_1} = \arg\min ||Ax - b||_1$

$\ell_1$-norm distribution is **wider** with a **high peak at zero**
$l_\infty$-norm (Chebyshev) approximation

minimize $\|Ax - b\|_\infty$

The $\infty$-norm of $m$-vector $y$ is

$$\|y\|_\infty = \max_{i=1,\ldots,m} |y_i| = \max_{i=1,\ldots,m} \max\{y_i, -y_i\}$$

Equivalent problem

minimize $t$
subject to $-t1 \leq Ax - b \leq t1$

Matrix notation

minimize

\[
\begin{bmatrix}
0 \\
1
\end{bmatrix}^T \begin{bmatrix}
x \\
t
\end{bmatrix}
\]

subject to

\[
\begin{bmatrix}
A & -1 \\
-A & -1
\end{bmatrix} \begin{bmatrix}
x \\
t
\end{bmatrix} \leq \begin{bmatrix}
b \\
-b
\end{bmatrix}
\]
Sparse signal recovery via $\ell_1$-norm minimization

$\hat{x} \in \mathbb{R}^n$ is unknown signal, known to be sparse
We make linear measurements $y = A\hat{x}$ with $A \in \mathbb{R}^{m \times n}, m < n$

Estimate signal with smallest $\ell_1$-norm, consistent with measurements

$$\text{minimize} \quad \|x\|_1$$
$$\text{subject to} \quad Ax = y$$

Equivalent linear optimization

$$\text{minimize} \quad 1^T u$$
$$\text{subject to} \quad -u \leq x \leq u$$
$$Ax = y$$
Sparse signal recovery via $\ell_1$-norm minimization

**Example**

Exact signal $\hat{x} \in \mathbb{R}^{1000}$
10 nonzero components
Random $A \in \mathbb{R}^{100 \times 1000}$

The least squares estimate cannot recover the sparse signal

$$\text{minimize} \quad \|x\|_2^2$$
$$\text{subject to} \quad Ax = y$$

The $\ell_1$-norm estimate is exact

$$\text{minimize} \quad \|x\|_1$$
$$\text{subject to} \quad Ax = y$$
Sparse signal recovery via $\ell_1$-norm minimization

Exact recovery

When are these two problems equivalent?

\[
\begin{align*}
\text{minimize} & \quad \text{card}(x) \\
\text{subject to} & \quad Ax = y
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \|x\|_1 \\
\text{subject to} & \quad Ax = y
\end{align*}
\]

\(\text{card}(x)\) is cardinality (number of nonzero components) of \(x\)

We say \(A\) allows **exact recovery** of \(k\)-sparse vectors if

\[
\hat{x} = \arg\min_{Ax=y} \|x\|_1 \quad \text{when} \ y = A\hat{x} \text{ and } \text{card}(\hat{x}) \leq k
\]

It depends on the nullspace\(^1\) of the “measurement matrix” \(A\)

---

1. Feuer & Nemirovski (IEEE Trans. On Information Technology, 2003) and several other papers on compressed sensing.
Linear classification
Support vector machine (linear separation)

Given a set of points \( \{v_1, \ldots, v_N\} \) with binary labels \( s_i \in \{-1, 1\} \)
Find hyperplane that strictly separates the two classes

\[
a^T v_i + b > 0 \quad \text{if} \quad s_i = 1
\]
\[
a^T v_i + b < 0 \quad \text{if} \quad s_i = -1
\]

Homogeneous in \((a, b)\), hence equivalent to the linear inequalities (in \(a, b\))

\[
s_i (a^T v_i + b) \geq 1
\]
Linear classification
Separable case

Feasibility problem

\[
\begin{align*}
\text{find} & \quad a, b \\
\text{subject to} & \quad s_i(a^T v_i + b) \geq 1, \quad i = 1, \ldots, N
\end{align*}
\]

Which can be seen as a special case of LP with

\[
\begin{align*}
\text{minimize} & \quad 0 \\
\text{subject to} & \quad s_i(a^T v_i + b) \geq 1, \quad i = 1, \ldots, N
\end{align*}
\]

\( p^* = 0 \) if problem feasible (points separable)

\( p^* = \infty \) if problem infeasible (points not separable)  \[ \rightarrow \text{What then?} \]
Linear classification

Approximate linear separation of non-separable points

\[
\text{minimize } \sum_{i=1}^{N} (1 - s_i(a^T v_i + b))_+ = \sum_{i=1}^{N} \max\{0, 1 - s_i(a^T v_i + b)\}
\]

If \(v_i\) misclassified, \(1 - s_i(a^T v_i + b)\) is the penalty

Piecewise-linear minimization problem with variables \(a, b\)
Linear classification
Approximate linear separation of non-separable points

minimize $\sum_{i=1}^{N} \max\{0, 1 - s_i(a^Tv_i + b)\}$

Equivalent problem

minimize $\sum_{i=1}^{N} u_i$
subject to $1 - s_i(v_i^Ta + b) \leq u_i, \quad i = 1, \ldots, N$
$0 \leq u_i, \quad i = 1, \ldots, N$

Matrix notation?
Modelling software for linear programs

Modelling tools simplify the formulation of LPs (and other problems)

- Accept optimization problem in common notation ($\max, \| \cdot \|_1, \ldots$)
- Recognize problems that can be converted to LPs
- Express the problem in input format required by a specific LP solver

Examples
- AMPL, GAMS
- CVX, YALMIP (Matlab)
- CVXPY, Pyomo (Python)
- JuMP.jl, Convex.jl (Julia)
**CVXPY example**

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|_1 \\
\text{subject to} & \quad 0 \leq x \leq 1
\end{align*}
\]

```python
x = cp.Variable(n)
objective = cp.Minimize(cp.norm(A*x - b, 1))
constraints = [0 <= x, x <= 1]
problem = cp.Problem(objective, constraints)

# The optimal objective value is returned by `problem.solve()`.
result = problem.solve()

# The optimal value for x is stored in `x.value`.
print(x.value)
```
Why linear optimization?

“Easy” to solve

- It is solvable in polynomial time, and it is tractable in practice
- State-of-the-art software can solve LPs with tens of thousands of variables. We can solve LPs with millions of variables with specific structure.

Extremely versatile

It can model many real-world problems, either exactly or approximately.

Fundamental

The theory of linear optimization lays the foundation for most optimization theories
Next lecture
Geometry of linear optimization

• Polyhedra
• Extreme points
• Basic feasible solutions