# Lecture 1: Modelling and classification

# Optimization

"Optimum:" Latin for "the ultimate ideal;" similarly, "optimus:" "the best." To optimize is to bring something to its ultimate state.

**Example problem**: Consider a hospital ward which operates 24 hours a day. At different times of day, the staff requirement differs. Table 1 shows the demand for reserve wardens during six work shifts.

| **Shift** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Hours** | 0–4 | 4–8 | 8–12 | 12–16 | 16–20 | 20–24 |
| **Demand** | 8 | 10 | 12 | 10 | 8 | 6 |

Table 1: Staff requirements at a hospital ward

Each member of staff works in 8 hour shifts. The goal is to fulfill the demand with the least total number of reserve wardens.

# A staff planning problem

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}}\ \ & f(\boldsymbol{x}) := \sum_{j=1}^{6} x_j, \\
\text{subject to}\ \ & x_6 + x_1 \geq 8, && \text{(work ends at shift 1)} \\
& x_1 + x_2 \geq 10, \\
& x_2 + x_3 \geq 12, \\
& x_3 + x_4 \geq 10, \\
& x_4 + x_5 \geq 8, \\
& x_5 + x_6 \geq 6, && \text{(work ends at shift 6)} \\
& \qquad\quad x_j \geq 0, && j = 1, \dots, 6, \\
& \qquad\quad x_j \text{ integer}, && j = 1, \dots, 6.
\end{aligned}
$$

*Optimal solution*: $\boldsymbol{x}^*$, a vector of decision variable values which gives the objective function its minimal value among the feasible solutions.

Two optimal solutions: $\boldsymbol{x}^* = (4, 6, 6, 4, 4, 4)^{\mathrm{T}}$, $\boldsymbol{x}^* = (8, 2, 10, 0, 8, 0)^{\mathrm{T}}$

*Optimal value*: $f(\boldsymbol{x}^*) = 28$.

The above model is a crude simplification of any real application.

Should add requirements on individual competence, more detailed restrictions, longer planning horizon, employment rules etcetera.

More complex models in practice.

# Modelling practice

Figure 1 illustrates several issues in the modelling process.

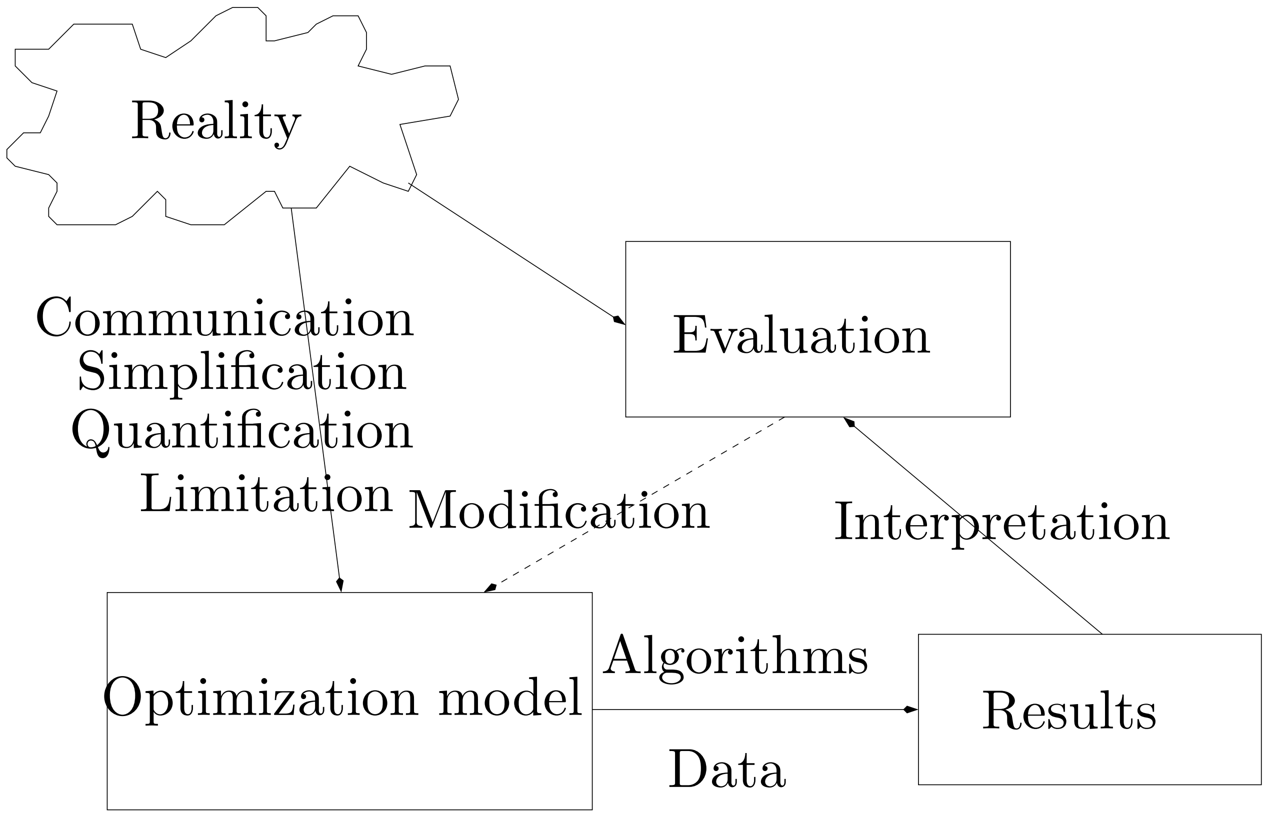



Figure 1: Flow chart of the modelling process

## Difficulties

- Communication can often be difficult (the two parties speak different languages in terms of describing the problem)
- Problems with data collection:
  - Quantification difficult
  - Enough accuracy obtained?
  - Uncertainties (sometimes part of the problem, sometimes not)
- Conflict between problem solvability and problem realism
- Problems with the result:
  - Interpretation of the result must make sense to users
  - Must be possible to transfer the solution back into the "fluffy" world where the problem came from

# Problem classification, I: General problem

$$
\begin{aligned}
\boldsymbol{x} \in \mathbb{R}^n &: \text{vector of decision variables } x_j, \quad j = 1, 2, \ldots, n; \\
f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{\pm\infty\} &: \text{objective function;} \\
X \subseteq \mathbb{R}^n &: \text{ground set defined logically/physically;} \\
g_i : \mathbb{R}^n \mapsto \mathbb{R} &: \text{constraint function defining restriction on } \boldsymbol{x} :
\end{aligned}
$$

$$
\begin{aligned}
g_i(\boldsymbol{x}) &\geq 0, \qquad i \in \mathcal{I}; \quad \text{(inequality constraints)} \\
g_i(\boldsymbol{x}) &= 0, \qquad i \in \mathcal{E}. \quad \text{(equality constraints)}
\end{aligned}
$$

# Problem classification, I: General problem

The optimization problem then is to

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}), \\
\text{subject to} \quad & g_i(\boldsymbol{x}) \geq 0, \qquad i \in \mathcal{I}, \\
& g_i(\boldsymbol{x}) = 0, \qquad i \in \mathcal{E}, \\
& \boldsymbol{x} \in X.
\end{aligned}
$$

(If it is really a maximization problem, then we change the sign of $f$.)

## Example problems

**(LP) Linear programming** Objective function linear:
$f(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = \sum_{j=1}^{n} c_j x_j$ ($\boldsymbol{c} \in \mathbb{R}^n$);
constraint functions affine: $g_i(\boldsymbol{x}) = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i$ ($\boldsymbol{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $i \in \mathcal{I} \cup \mathcal{E}$);
$X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid x_j \geq 0, \quad j = 1, 2, \ldots, n \,\}$.

**(NLP) Nonlinear programming** Some function(s) $f, g_i$ ($i \in \mathcal{I} \cup \mathcal{E}$) are nonlinear.

**Continuous optimization** $f, g_i$ $(i \in \mathcal{I} \cup \mathcal{E})$ are continuous on an open set containing $X$;
$X$ is closed and convex.

**Integer programming** $X \subseteq \{0, 1\}^n$ or $X \subseteq \mathbb{Z}^n$.

**Unconstrained optimization** $\mathcal{I} \cup \mathcal{E} = \emptyset$;
$X = \mathbb{R}^n$.

**Constrained optimization** $\mathcal{I} \cup \mathcal{E} \neq \emptyset$ and/or $X \subset \mathbb{R}^n$.

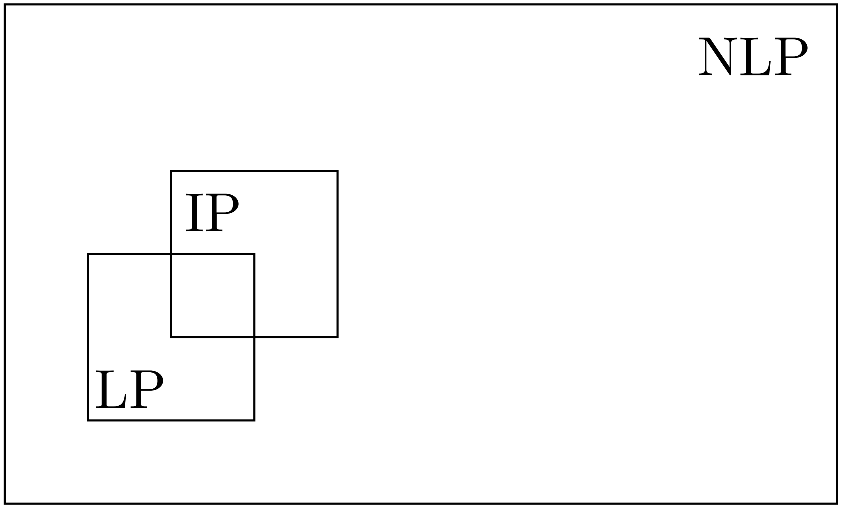
**Differentiable optimization** $f, g_i$ $(i \in \mathcal{I} \cup \mathcal{E})$ are at least once continuously differentiable on an open set containing $X$ (that is, "in $C^1$ on $X$," which means that $\nabla f$ and $\nabla g_i$ exist there and the gradients are continuous);
further, $X$ is closed and convex.

**Non-differentiable optimization** At least one of $f, g_i$ $(i \in \mathcal{I} \cup \mathcal{E})$ is non-differentiable.

**(CP) Convex programming** $f$ is convex; $g_i$ $(i \in \mathcal{I})$ are concave;
$g_i$ $(i \in \mathcal{E})$ are affine;
$X$ is closed and convex.

**Non-convex programming** The complement of the above

Relations among NLP, IP, and LP:




LP special case of NLP: a linear function is a special kind of nonlinear function (cf. Taylor expansion)

IP special case of NLP: $x_j \in \{0, 1\}$ equivalent to $x_j(1 - x_j) = 0$

Some IP problems are equivalent to LP—*integrality property*. (Example: The shortest path problem)

## Rough distinctions between LP and NLP

LP Linear programming $\approx$ applied linear algebra. LP is "easy," because there exist algorithms that can solve every LP problem instance efficiently in practice.

NLP Nonlinear programming $\approx$ applied analysis in several variables. NLP is "hard," because there does *not* exist an algorithm that can solve every NLP problem instance efficiently in practice. NLP is such a large problem area that it contains very hard problems as well as very easy problems. The largest class of NLP problems that are solvable with some algorithm in reasonable time is CP (of which LP is a special case).

## What then is optimization?

- If there are no $\geq$- or $\leq$-constraints then the problem is essentially unconstrained.
- $=$-constraints are treated through numerical analysis techniques. So, unconstrained optimization is essentially a numerical analysis subject.
- With $\geq$- or $\leq$-constraints we face problems such as which are the active constraints. One-sidedness.
- Results in difficult “non-differentiabilities.”
- Largely a subject of convex and variational analysis. This is optimization!