Lecture 1: Modelling, classification, and a road map of the course

Michael Patriksson

20 January 2003

Optimization

“Optimum:” latin for “the ultimate ideal;” similarly, “optimus:” “the best.” To optimize is to bring something towards 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

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

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

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

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

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

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.

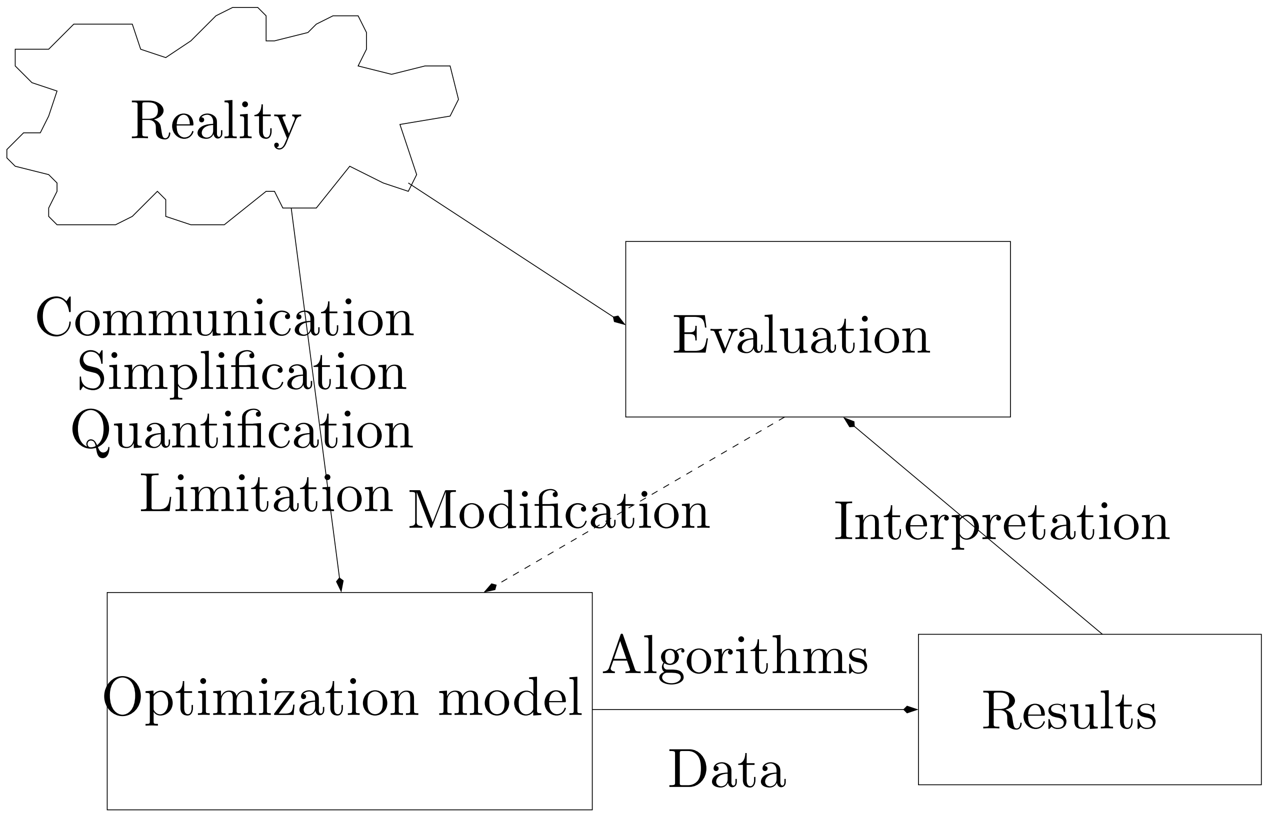


Figure 1: Flow chart of the modelling process

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

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

(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:
$g_i(\boldsymbol{x}) = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i$ ($\boldsymbol{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$);
$X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid x_j \geq 0, \quad j = 1, 2, \ldots, n \,\}$.

(NLP) Nonlinear programming Some function(s) f, g_i are nonlinear.

Continuous optimization f, g_i are continuous on an open set containing X; X is closed and convex.

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

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

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

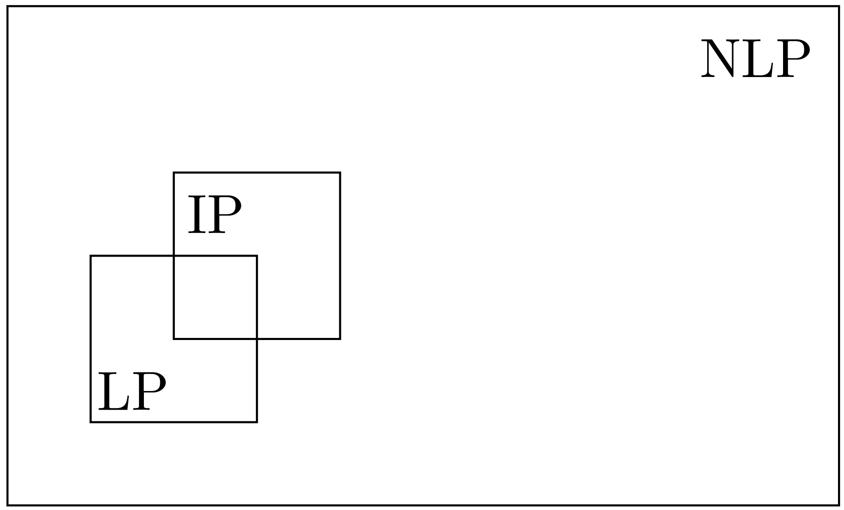
Differentiable optimization f, g_i are at least once continuously differentiable on an open set containing X (that is, "in C^1 on X," which means that ∇f and ∇g_i exist there and the gradients are continuous); further, X is closed and convex.

Non-differentiable optimization At least one of f, g_i 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; and 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

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

NLP Nonlinear programming $\sim$ 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!

Course map

Chapter 3—Convexity analysis

- Utilized in every chapter!
- Introduces convex sets and functions, relates the two. Characterizes convex functions in several variables [past knowledge in $\mathbb{R}$: $f \in C^2$ convex $\iff f''(x) \geq 0$ everywhere]
- Important special case: polyhedral sets (formed by linear constraints). Theorems: Representation Theorem; Separation Theorem; Farkas' Lemma. Foundation of LP.

Chapter 4—Optimality conditions, convex sets

- How do we know if a point is globally or locally optimal?
- Past knowledge in $\mathbb{R}$: Check boundary points, $f'(x) = 0$ (stationary points), discontinuities in f or f'. Less useful in $\mathbb{R}^n$.
- Theorem: Local minima are global for convex problems.
- Existence of solutions [Past knowledge: Weierstrass' Theorem].
- Covers C^1 and C^2 cases, unconstrained case. Basic story: $\nabla f(\boldsymbol{x}^*) = \mathbf{0}^n$, psd. Hessian; no guarantees without convexity.
- Constrained optimization. Local optimality at $\boldsymbol{x}^* \Longrightarrow \nexists$ feasible directions from $\boldsymbol{x}^*$ which yield descent; several characterizations.
- For convex sets, less technical. For general sets: the Karush–Kuhn–Tucker conditions (Chapter 6). More technical, requires "dual." This chapter is "primal."

Chapter 5—Unconstrained optimization methods

- Iterative algorithm: [1] find descent direction (approximate the original problem); [2] perform a line search (solve original problem over a line segment); [3] Update; [4] Check termination criterion, stop or return to [1].
- Steepest descent (C^1); Newton-type methods (C^1 or C^2).
- General convergence results; convergence rates.
- What happens in the non-differentiable case?
- Modern methods: trust-region, pattern search.

Chapter 6—Optimality conditions, general case

- More difficult to define “feasible directions” and such sets.
- Must introduce regularity conditions to be able to state anything interesting. “Constraint qualifications” (CQ).
- Important to distinguish between active/inactive constraints.
- “Primal–dual” conditions—introduce “Lagrange multipliers,” zero for inactive constraints. Sign restricted for $\leq/\geq$.
- Theorems: Fritz-John Theorem; Karush–Kuhn–Tucker Theorem. Utilizes Farkas’ Lemma etc. from Chapter 3.
- Natural geometric conditions as a “force equilibrium” in a fight between f and the active constraints.
- The responsibility always lies with the user!
- Convexity $\Longrightarrow$ works fine. KKT points are globally optimal.

Chapter 7—Lagrangian duality

- Relaxation: larger set and/or underestimating function $\Longrightarrow$ easier problem. Ex: linearization of f. Lagrange relaxation of constraints, larger set, penalize violation with Lagrangian terms.
- Find best lower bound by solving a Lagrangian dual problem in the multipliers.
- If convex the problems (primal and dual) have same optimal value: Strong Duality Theorem. Can use dual solution to translate into primal optimum. [Past knowledge: "Lagrange multiplier method" for simple =-constrained problems] Otherwise: duality gap, no easy translation.
- Strong duality utilizes convexity: Separation Theorem, CQ.
- Dual problem convex but often non-differentiable. Subgradients replace gradients, convergent methods.

Chapter 8–11—Linear programming

- Special methods can be used: find best extreme point by moving between neighbours in the polyhedron. Simplex method.
- Extremely efficient in practice, not in worst-case theory.
- LP special case where CQ holds—strong duality prseent, KKT necessary and sufficient. Also: existence of solutions easily characterized.
- Theorems: Representation Theorem, Strong Duality, Complementarity.
- KKT multipliers = Lagrange multipliers = Dual variables.
- Sensitivity analysis: what happens to f^*, $\boldsymbol{x}^*$ if we change data?

Chapter 12—Convexly constrained optimization

- Linearization of f yields LP—easy to construct feasible descent methods if S polyhedral.
- Frank–Wolfe method utilizes LP + line searches. Utilizes Relaxation Theorem; optimality conditions from Chapter 4.
- Simplicial decomposition: Store extreme points generated, optimize over their convex hull. Utilizes Representation Theorem. Smarter than FW.
- Gradient projection method. Utilizes characterization of stationary point by means of the projection of a perturbed point (Chapter 4). Solves QP problems to get descent.
- Every iterative method utilizes relaxation of the optimality conditions and/or the original problem!

Chapter 13—General constrained optimization

- Relaxation by penalizing constraints: penalty methods.
- Exterior penalty: add positive function of constraint violation—”stronger” than Lagrangian relaxation. Higher penalty parameter enforces feasibility. Infeasible method.
- Interior penalty for convex sets: add interior asymptotes for the boundaries. Smaller penalty parameters allow boundaries to be approached if needed. Strictly feasible method. Interior point method for LP polynomial! Better than Simplex in theory and sometimes in practice.
- Relaxation by means of linearizing the KKT conditions $\Longrightarrow$ Sequential Quadratic Programming (SQP). $\sim$ Newton’s method for KKT system + penalty function; modifications necessary with $\leq/\geq$-constraints. Filter-SQP: no penalty functions.