Introduction and Preliminaries

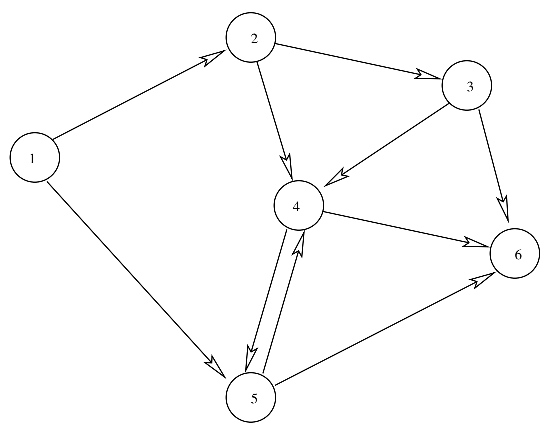
Lecture 1, Continuous Optimisation

Oxford University Computing Laboratory, HT 2006

Notes by Dr Raphael Hauser (hauser@comlab.ox.ac.uk)

Example 1: Linear Programming

A network of gas pipelines is given.



- An arrow from node i to node j represents a pipe with transport capacity c_{ij} in the given direction.

- Transporting one unit of gas along the edge (ij) costs d_{ij}.

- The amount of gas produced at node i is p_i,

- and the amount of gas consumed is q_i.

- We assume that the total amount consumed equals the total amount of gas produced.

- How to choose the quantities x_{ij} of gas shipped along the edges (ij) so as to minimise costs while satisfying demands?

We set $c_{ij} = 0$ (and d_{ij} arbitrary numbers) for all edges (ij) that do not exist. Doing so, we can assume that the network is a complete graph.

The problem we have to solve is the following:

$$\min_x \sum_{i,j=1}^{6} d_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{k=1}^{6} x_{ki} + p_i = \sum_{j=1}^{6} x_{ij} + q_i, \qquad (i = 1, \ldots, 6), \tag{1}$$

$$0 \le x_{ij} \le c_{ij}, \qquad (i, j = 1, \ldots, 6). \tag{2}$$

- This is an example of a *linear programming* problem, as the *objective function* $\sum_{i,j=1}^{6} d_{ij}x_{ij}$ and the *constraint functions* (1),(2) are linear functions of the *decision variables* x_{ij}.

- Note that it is not a priori clear that this problem has feasible solutions. One is therefore interested in algorithms that not only find optimal LP solutions when these exist but also detect when a problem instance is infeasible!

- Furthermore, if there is an optimal solution, we are not only interested in the minimum value of the objective function, but also in the values of x_{ij} that achieve this minimum. Such an x is called a *minimiser* of the problem.

Example 2: Quadratic Programming

- An investor considers a fixed time interval and wishes to decide which fraction of the capital he/she wants to invest in each of n different given assets.

- The expected return of asset i is μ_i, assumed known.

- The covariance between assets i and j is σ_{ij}, assumed known.

- The investor aims at a total return of at least b.

- Subject to this constraint, how to minimise the variance of the overall portfolio (notion of risk)?

This problem can be modelled as

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \\
\text{s.t.} \quad & \sum_{i=1}^{n} \mu_i x_i \geq b, \\
& \sum_{i=1}^{n} x_i = 1, \\
& x_i \geq 0 \quad (i = 1, \dots, n).
\end{aligned}
$$

The constraint $\sum_{i=1}^{n} x_i = 1$ expresses the requirement that 100% of the initial capital has to be invested.

Example 3: Semidefinite Programming

- In optimal control, variables $y_1, \ldots, y_m$ have to be chosen so as to design a system that is driven by the linear ODE

$$\dot{u} = M(y)u,$$

  where $M(y) = \sum_{i=1}^{m} y_i A_i + A_0$ is an affine combination of given symmetric matrices A_i $(i = 0, \ldots, m)$.

- To stabilise the system, one would like to choose y so as to minimise the largest eigenvalue of $M(y)$.

Note that $\lambda_1(M) \leq \eta$ if and only if $\eta\,\mathrm{I} - M$ has only non-negative eigenvalues.

This is equivalent to $\eta\,\mathrm{I} - M$ being positive semidefinite, denoted by $\eta\,\mathrm{I} - M \succeq 0$.

The problem we need to solve is thus the following,

$$\begin{aligned} &\max_{\eta, y} \; -\eta \\ &\text{s.t.} \quad \eta\,\mathrm{I} - A_0 - \sum_{i=1}^{m} y_i A_i \succeq 0. \end{aligned}$$

Example 4: Polynomial Programming

- An engineer designs a system determined by two design variables x and y which are dependent on each other via the relation $xy = 1$.

- The energy consumed by the system is given by $E(x, y) = x^2 + y^2 - 4$.

- The physical properties of materials used impose the constraints $x \in [0.5, 3]$.

- How to design a system that consumes the smallest amount of energy among all admissible systems?

This problem can be formulated as

$$
\begin{aligned}
(\mathrm{P}) \qquad \min_{x,y}\ & x^2 + y^2 - 4 \\
\text{s.t.}\quad & x - 0.5 \geq 0, \\
& -x + 3 \geq 0, \\
& xy - 1 = 0.
\end{aligned}
$$

The General Problem:

More generally, a *continuous programming problem* concerns the minimisation (or maximisation) of a continuous objective function f under constraints defined by continuous functions g_i, h_j:

$$
\begin{aligned}
\text{(P)} \quad & \min_{x \in \mathbb{R}^n} f(x) \\
& \text{s.t. } g_i(x) \geq 0 \quad (i = 1, \dots, p) \\
& \qquad h_j(x) = 0 \quad (j = 1, \dots, q).
\end{aligned}
$$

- Typically we will assume $f, g_i, h_j \in C^2$.
- $g_i(x) \geq 0$ are called *inequality constraints*.
- $h_j(x) = 0$ are called *equality constraints*.
- Constraints of the form $x_i \in \mathbb{Z}$ (integrality constraints) add a whole other layer of difficulty we will not consider in this course (see Section B course Integer Programming).

What are key properties of iterative algorithms?

- Correct termination: does the algorithm converge to a minimiser? ($\rightarrow$ to recognise optima, need to characterise them mathematically, i.e., develop *optimality conditions*).

- Low complexity:
  i. low total number of iterations ($\rightarrow$ need a notion of *convergence rate*),
  ii. low number of computer operations *per* iteration ($\rightarrow$ often leads to a trade-off with i.).

- Reliability: how sensitive is the algorithm to small changes in input, how is it affected by round-off?

Some Terminology:

- $x \in \mathbb{R}^n$ is called *feasible solution* for (P) if it satisfies all the constraints, that is, if

$$g_i(x) \geq 0 \quad (i = 1, \dots, p),$$
$$h_j(x) = 0 \quad (j = 1, \dots, q).$$

- The set of feasible solutions is denoted by $\mathcal{F}$, also called the feasible set. Hence, (P) can be formulated as

$$\min\{f(x) : x \in \mathcal{F}\}.$$

- $x \in \mathbb{R}^n$ is *strictly* feasible if

$$
\begin{aligned}
g_i(x) > 0 \quad (i = 1, \ldots, p), \\
h_j(x) = 0 \quad (j = 1, \ldots, q).
\end{aligned}
$$

- The set of strictly feasible solutions is denoted by $\mathcal{F}^\circ$. This is the relative interior of $\mathcal{F}$.

- $x \in \mathcal{F}$ is a *local minimiser* of (P) if there exists $\varepsilon > 0$ such that

$$f(x^*) \leq f(x) \quad \forall\, x \in \mathcal{F} \cap B_\varepsilon(x^*).$$

- $x^* \in \mathcal{F}$ is a *global minimiser* of (P) if

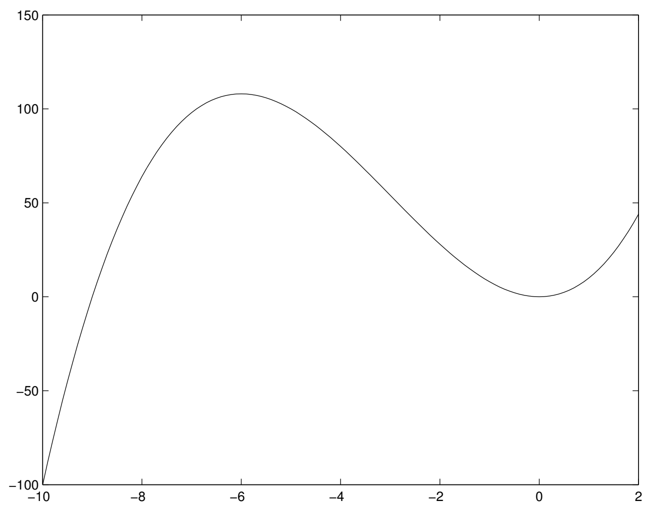
$$f(x^*) \leq f(x) \quad \forall\, x \in \mathcal{F}.$$

Example 5: Local versus global optimisation

The problem

$$(P) \qquad \min_{x\in\mathbb{R}} f(x) = x^3 + 9x^2$$
$$\text{s.}t. \quad -10 \le x \le 2$$

has a local minimiser at $x = 0$, and a global minimiser at $x^* = -10$.



Q-linear convergence:

- A sequence $(x_k)_{\mathbb{N}} \to x^* \in \mathbb{R}^n$ converges Q-linearly if there exists $\rho \in (0,1)$ (the *convergence factor*) and $k_0 \in \mathbb{N}$ such that

$$\|x_{k+1} - x^*\| \leq \rho \|x_k - x^*\| \quad \forall\, k \geq k_0.$$

- Practical significance: x_k approximates x^* to $O(-\log_{10}\|x_k - x^*\|)$ correct digits. Therefore, $O(-log_{10}\rho)$ additional correct digits appear per iteration:

$$-\log_{10}\|x_{k+1} - x^*\| - \left(-\log_{10}\|x_k - x^*\|\right) \geq -\log_{10}\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \simeq -\log \rho.$$

Example 6:

Let $z \in (0,1)$ be fixed and consider the sequence $(x_k)_\mathbb{N}$ of k-th partial geometric series

$$x_k = \sum_{n=0}^{k} z^n.$$

Then $(x_k)_\mathbb{N}$ converges to $x^* = \frac{1}{1-z} \in \mathbb{R}^1$ Q-linearly with $\rho = z$: for all k,

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|} = \frac{\sum_{n=k+2}^{\infty} z^n}{\sum_{m=k+1}^{\infty} z^m} = z < 1.$$

Q-superlinear convergence:

- A sequence $(x_k)_{\mathbb{N}} \to x^* \in \mathbb{R}^n$ converges Q-superlinearly if

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

- Faster than linear for all ρ.

- Practical significance: asymptotically, the number of additional correct digits per iteration becomes larger than any fixed number.

Q-convergence of rate $r > 1$:

- A sequence $(x_k)_{\mathbb{N}} \to x^* \in \mathbb{R}^n$ converges at the Q-rate $r > 1$ if there exists k_0 such that

$$\|x_{k+1} - x^*\| \leq \|x_k - x^*\|^r \quad \forall\, k \geq k_0.$$

- Practical significance: the number of additional correct digits is approximately multiplied by r in each iteration:

$$-\log_{10} \|x_{k+1} - x^*\| \simeq r\Big(-\log_{10} \|x_k - x^*\|\Big).$$

Reading Assignment: Read up on convexity on pages 6–8 of Lecture-Note 1, which can be downloaded from the course web page.