46770 Integrated Energy Grids

Aleksander Grochowicz

Lecture 2 – Optimisation (a short review)

Balancing renewable generation

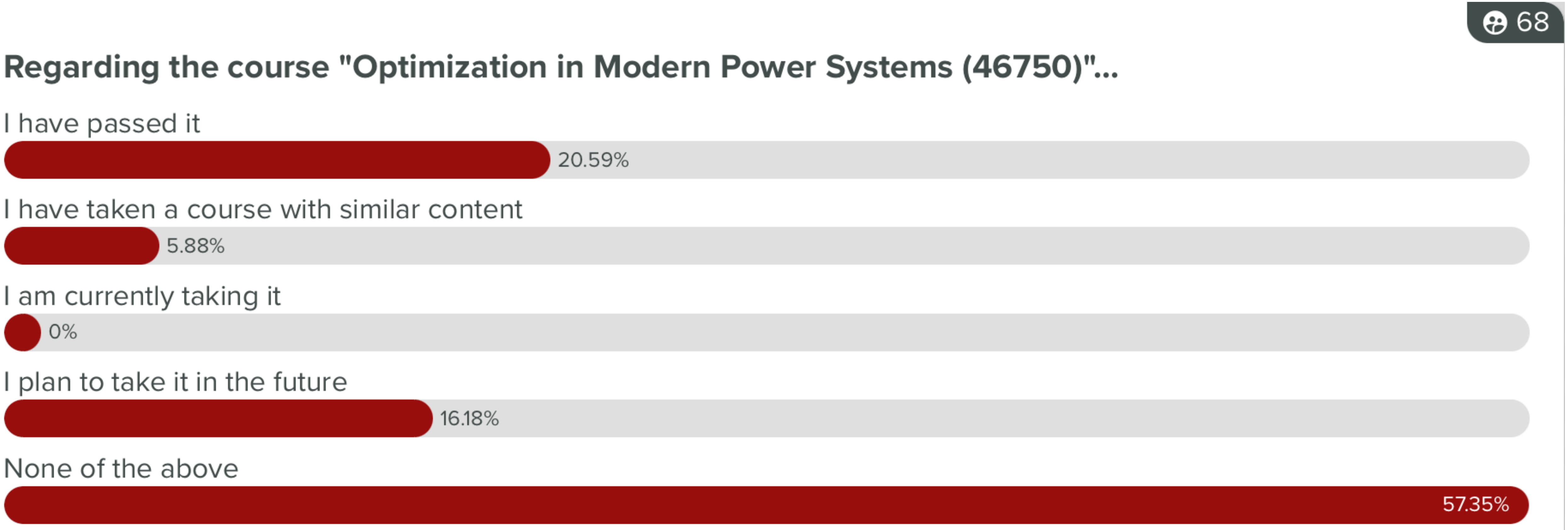
Last lecture: we need optimisation to decide how to combine different flexibility options.

1. Back-up generation and curtailment
2. Storage
3. Regional integration of renewables
4. Demand-side management
5. Sector coupling / other carriers

Temporal and spatial balancing (together with demand-side management and sector coupling) must be simultaneously considered ⇒ **This requires optimisation!**

This lecture provides a review of optimisation theory that we need in this course.

For a comprehensive discussion, we refer to the DTU course 46750 Optimisation in Modern Power Systems.


68
Regarding the course "Optimization in Modern Power Systems (46750)"...
I have passed it
20.59%
I have taken a course with similar content
5.88%
I am currently taking it
0%
I plan to take it in the future
16.18%
None of the above
57.35%

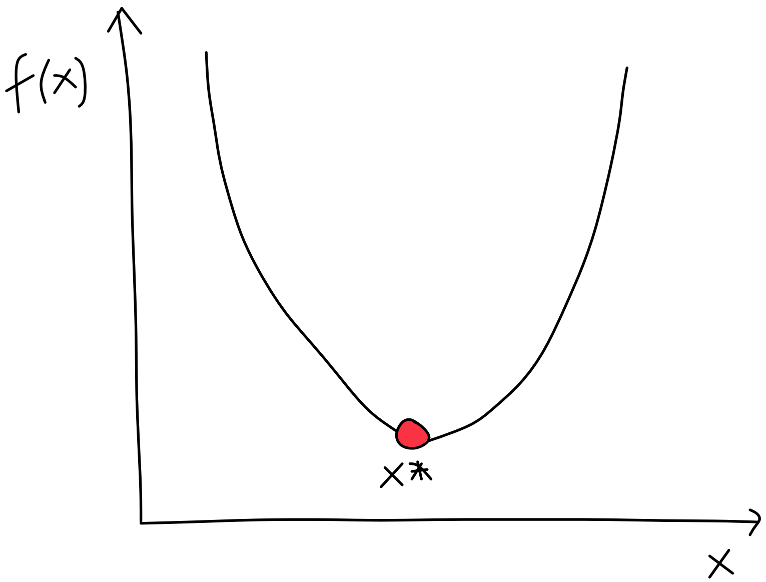
Learning goals for this lecture

- Formulate and solve optimisation problems subject to constraints
- Interpret the meaning of Lagrange/KKT multipliers associated with the constraints of an optimisation problem and analyse their values
- Formulate and solve simple economic dispatch problems
- Describe convex optimisation problems

Optimisation theory

One-dimension optimisation problem

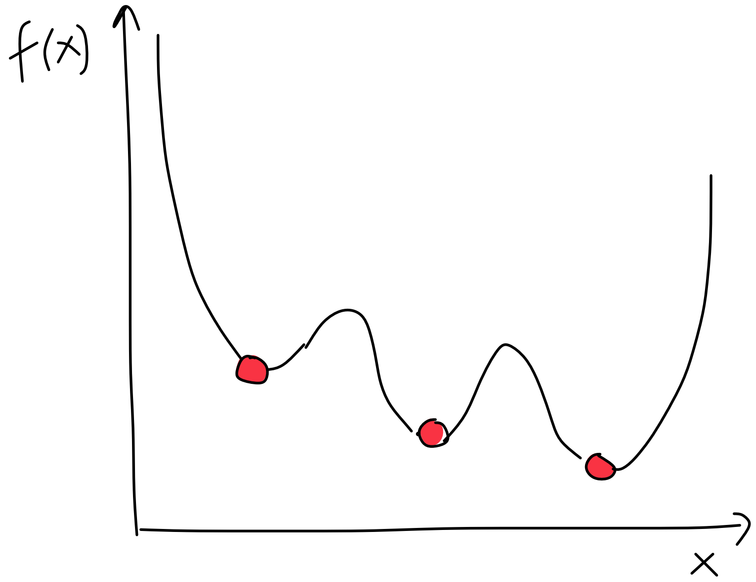
To calculate the optimum in one-dimensional problems, we calculate the derivative and make it equal to zero



$$\min_x f(x)$$

$$\left.\frac{\partial f}{\partial x}\right|_{x=x^*} = 0$$

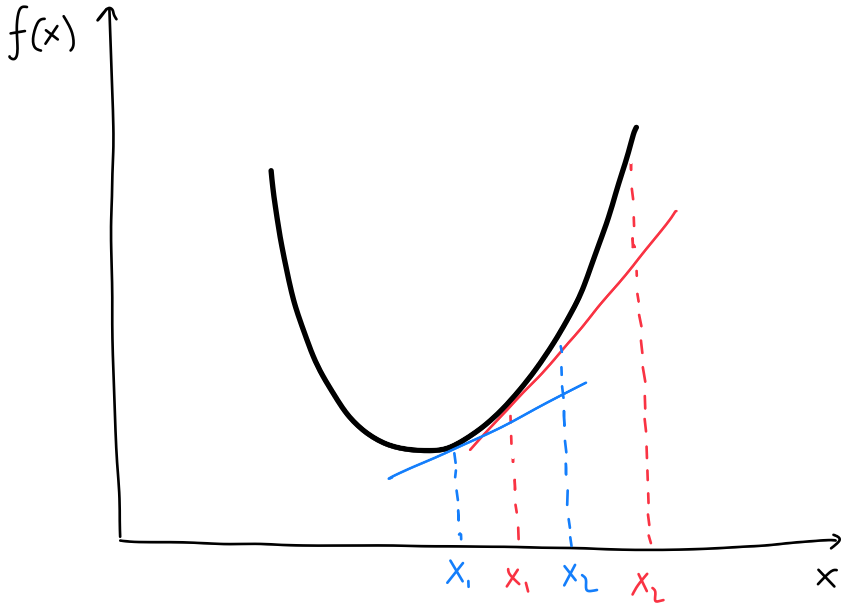
This method fails if the function is not convex



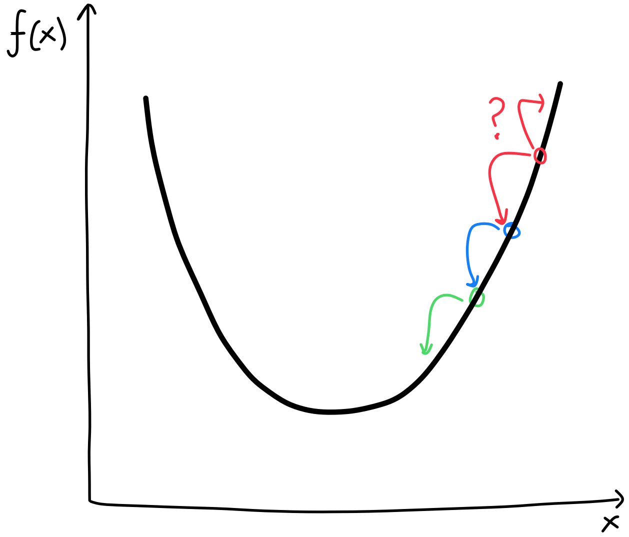
One-dimension optimisation problem

Convex function: $f(x)$ is above all its tangents, i.e.

$$\text{for all } x_1, x_2 \in \mathbb{R}^N : f(x_2) \geq (x_2 - x_1) \cdot \frac{\partial f(x_1)}{\partial x}$$



Convexity guarantees the convergence of gradient descent algorithms to a global optimum.



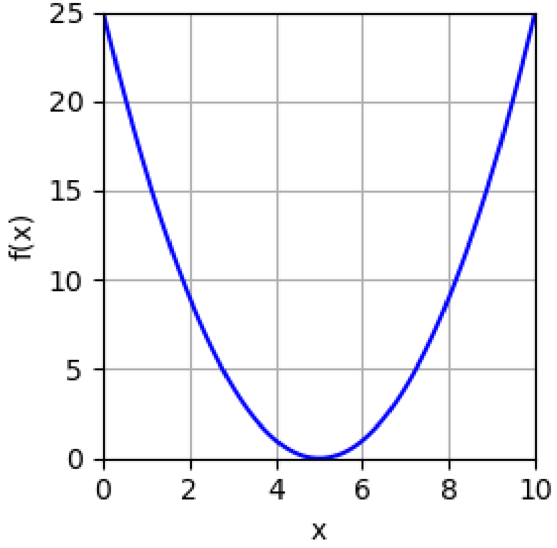
You can check out his nice video:

[Visually Explained: Newton's Method in optimisation]

Examples of convex and non-convex functions

Convex function:

$$f(x) = (x-5)^2$$



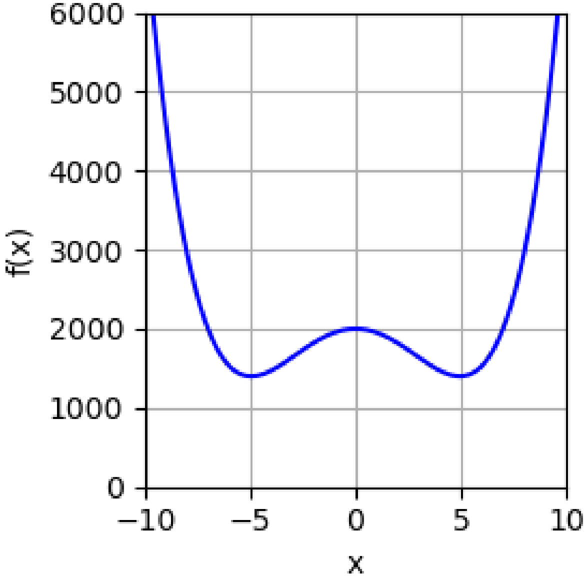
$$\frac{\partial f}{\partial x} = 2(x-5) = 0$$

$$x^* = 5$$

$$f(x^*) = 0$$

Non-convex function:

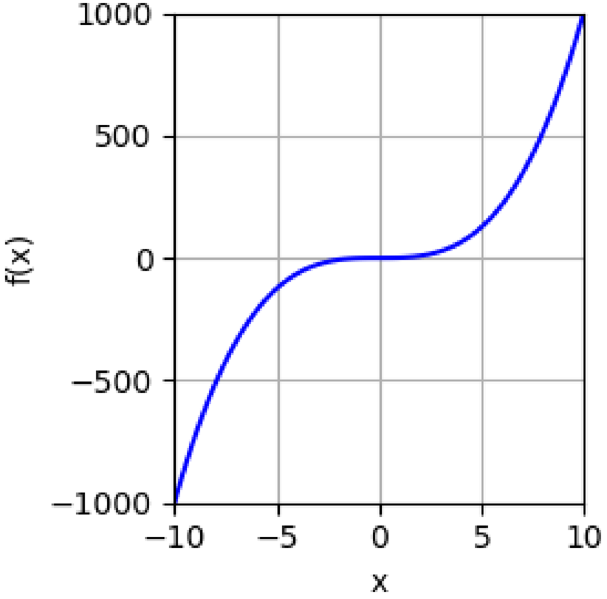
$$f(x) = x^4 - 7x^2 + 2000$$



Two local minima.
Algorithms that look for global minima do not work

Non-convex function:

$$f(x) = x^3$$

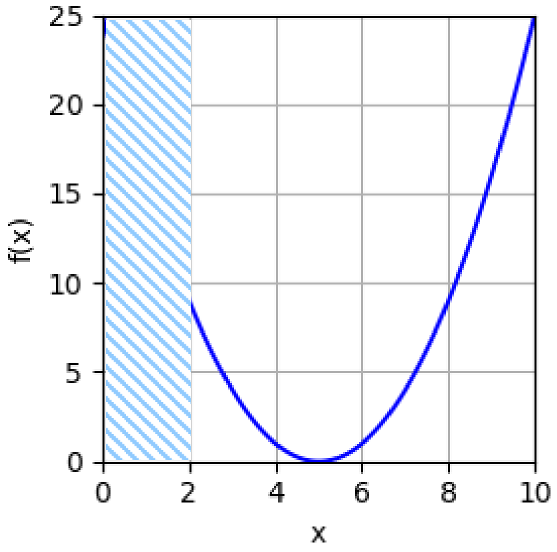


Derivative equals to zero indicates a saddle point.

And the function is not bounded

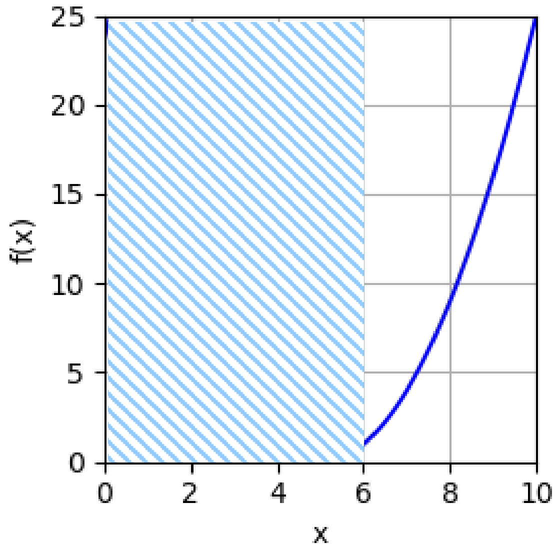
One-dimension optimisation problem with constraint

$$\begin{cases} \min f(x) = (x-5)^2 \\ \text{subject to:} \\ x \geq 2 \end{cases}$$



The constraint is **not binding**.
It has **no effect on the solution**.

$$\begin{cases} \min f(x) = (x-5)^2 \\ \text{subject to:} \\ x \geq 6 \end{cases}$$



The constraint is **binding**.
It modifies the solution

General formulation of optimisation problems (I)

$$\begin{cases} \min_{x} f(x) \\ \\ \text{subject to:} \\ \qquad h_i(x) = c_i \leftrightarrow \lambda_i \\ \qquad g_j(x) \geq d_j \leftrightarrow \mu_i \end{cases}$$

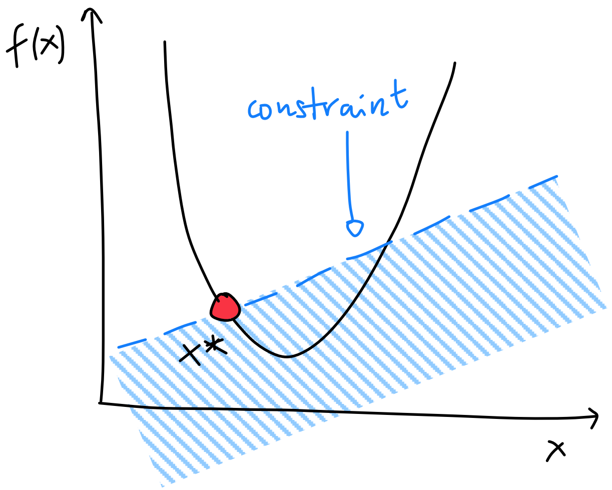
Objective function

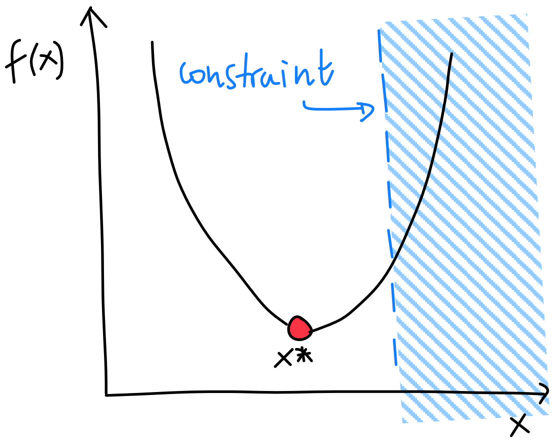
Equality and inequality constraints

λ_i and μ_j are the Lagrange or Karush-Kuhn-Tucker (KKT) multipliers

x, y, z are called primary variables and λ_i , μ_j are called dual variables

A constraint can be binding (affecting the optimal solution x^*) or not-binding





General formulation of optimisation problems (II)

In energy optimisation problems, the objective function typically attempts to minimise total costs, the constraints impose physical requirements (supply demand, maximum transmission capacity…) and the optimisation variables are the energy produced by every generator.

In this course, we will always build minimisation problems (for consistency, because we will follow the approach introduced in course 46750 Optimisation in Modern Power Systems).

Non-standard problem formulation

$$\begin{cases} \max_{x} f(x) & \text{Objective function} \\ \text{subject to:} & \\ \quad h_i(x) = c_i \leftrightarrow \lambda_i & \text{Equality and} \\ \quad g_j(x) \leq d_j \leftrightarrow \mu_j & \text{inequality constraints} \end{cases}$$

Standard problem formulation

$$\begin{cases} \min_{x} -f(x) \\ \text{subject to:} \\ \quad h_i(x) = c_i \leftrightarrow \lambda_i \\ \quad -g_j(x) + d_j \geq 0 \leftrightarrow \mu_j \end{cases}$$

See Problem 2.1

Using the Lagrangian to solve optimisation problems (I)

$$\min_x f(x)$$

Objective function

subject to:

$$h_i(x) - c_i = 0 \leftrightarrow \lambda_i$$

$$g_j(x) - d_j \geq 0 \leftrightarrow \mu_j$$

Equality and
inequality constraints

We can solve optimisation problems by building the Lagrangian function

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \sum_i \lambda_i [h_i(x) - c_i] - \sum_j \mu_j [g_j(x) - d_j]$$

and making its partial derivative with respect to x, λ, μ equal to zero.

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial f}{\partial x} - \sum_i \lambda_i \frac{\partial h_i}{\partial x} - \sum_j \mu_j \frac{\partial g_j}{\partial x} = 0 \qquad \frac{\partial \mathcal{L}}{\partial \lambda_i} = 0 \qquad \frac{\partial \mathcal{L}}{\partial \mu_j} = 0 \qquad \forall i, j$$

Using the Lagrangian to solve optimisation problems (II)

The Lagrange or Karush-Kuhn-Tucker (KKT) multipliers λ_i, μ_j represent the sensitivity of the optimal objective function value with respect to a small change in the right-hand side of the constraint c_i they are associated with $\lambda_i \sim \frac{\partial f}{\partial c_i}$ and $\mu_j \sim \frac{\partial f}{\partial d_j}$.

They have a relevant meaning in energy optimisation problems. They are also called **shadow prices**.

Here, we build the Lagrangian function with negative signs so that later we can obtain $\lambda_i \sim \frac{\partial f}{\partial c_i}$ but different $\pm$ criteria exist in the literature.

Example 1

Minimise the volume of a box subject to a constraint that limits the base perimeter plus length

$$\min_{x} f(x,y) = x^2 y$$

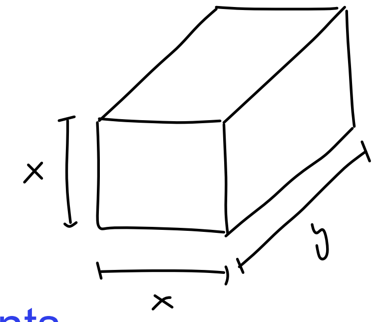
Objective function

subject to:

$$4x + y - 24 = 0 \leftrightarrow \lambda$$

$$x > 0$$

Equality and inequality constraints



To find a solution, we start by building the Lagrangian function

$$\mathcal{L}(x,\lambda,\mu) = f(x) - \sum_i \lambda_i [h_i(x) - c_i] - \sum_j [\mu_j [g_j(x) - d_j] = x^2 y - \lambda(4x + y - 24)$$

We derive the Lagrangian and make the derivative equal to zero

1) $$\frac{\partial \mathcal{L}}{\partial x} = 2xy - \lambda \cdot 4 = 0$$
$$\frac{\partial \mathcal{L}}{\partial y} = x^2 - \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 4x + y - 24 = 0.$$

2) $$4\lambda = 2xy$$
$$\lambda = x^2.$$

3) $$4x^2 = 2xy$$
$$2x = y$$
$$4x + 2x - 24 = 0.$$

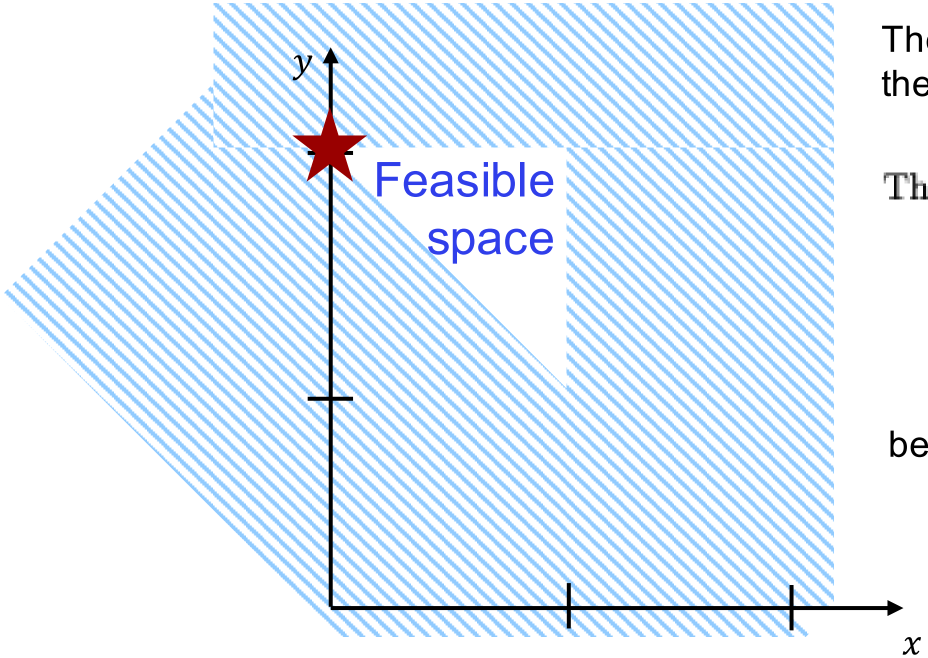
4) $$x^* = \frac{24}{6} = 4$$
$$y^* = 8$$
$$\lambda^* = 16.$$

Example 2

$$\min_{x,y} f(x) = 8x$$

subject to:

$$x + y \geq 2 \leftrightarrow \mu_1$$
$$-y + 2 \geq 0 \leftrightarrow \mu_2$$
$$-x + 1 \geq 0 \leftrightarrow \mu_3.$$



The constraints define the feasible space. If the space is empty, the problem is unfeasible.

The solution is $x^* = 0$

$y^* = 2$

$f^* = 0,$

and $\mu_3^* = 0,$

because the constraint is not binding.

To find a solution, we start by building the Lagrangian function

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \sum_i \lambda_i[h_i(x) - c_i] - \sum_j \mu_j[g_j(x) - d_j] = 8x - \mu_1(x + y - 2) - \mu_2(-y + 2) - \mu_3(-x + 1)$$

We derive the Lagrangian and make the derivative equal to zero

$$\frac{\partial \mathcal{L}}{\partial x} = 8 - \mu_1^* + \mu_3^* = 0 \qquad \mu_3^* = 0 \qquad \mu_1^* = 8$$

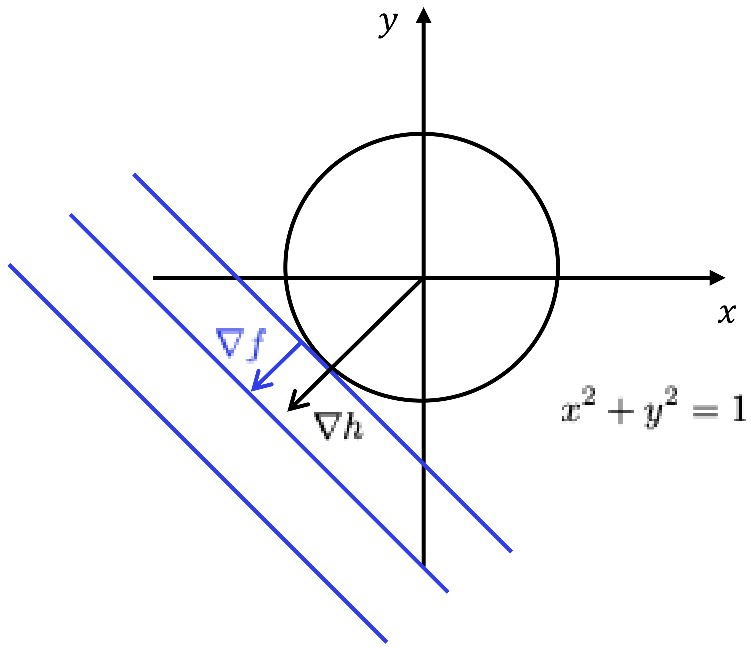
$$\frac{\partial \mathcal{L}}{\partial y} = 0 - \mu_1^* + \mu_2^* = 9 \qquad \mu_1^* = \mu_2^*.$$

$\mu_1^* = 8$ represents the change in the objective value of the optimal solution with respect to a small change in the constraint.

If we now assume $x + y \geq 2 - \varepsilon$,
the solution will be $x^* = -\varepsilon$.
$f^* = 8(-\varepsilon) = -8\varepsilon$ and $\frac{\partial f}{\partial d_j} = 8 \sim \mu_1^*$.

Example 3: Graphical interpretation of Lagrangian function

$$\begin{cases} \min_{x,y} f(x,y) = x + y \\ \text{subject to:} \\ \quad h(x,y) = x^2 + y^2 - 1 = 0 \end{cases}$$



At the point where the objective function is minimised and the constraint is fulfilled, the gradients of the objective function and the constrains are parallel $\nabla f = \lambda \nabla h$

$$\nabla f - \lambda \nabla h = 0$$

$$\frac{\partial \mathcal{L}}{\partial x} = 1 - 2\lambda x = 0 \qquad x = \frac{1}{2\lambda}$$

$$\frac{\partial \mathcal{L}}{\partial y} = 1 - 2\lambda y = 0 \qquad y = \frac{1}{2\lambda}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = x^2 + y^2 - 1 = 0 = (\frac{1}{2\lambda})^2 + (\frac{1}{2\lambda})^2 - 1 = 0$$

$$\lambda^* = \pm\frac{1}{\sqrt{2}} \qquad x^* = \pm\frac{\sqrt{2}}{2} \qquad y^* = \pm\frac{\sqrt{2}}{2}$$

Here, we can see again that the Lagrange multiplier corresponds to the sensitivity of the optimal objective function value with respect to a small change in the equality constraint they are associated with.

Sufficient and necessary Karush-Kuhn-Tucker (KKT) conditions

KKT are necessary conditions for an optimal solution.

In some cases, KKT are also sufficient conditions (e.g. linear optimisation problems)
KKT conditions are typically defined for maximisation problems, for a minimisation problem in the standard form that we use in this course, KKT conditions are the following

$$\frac{\partial \mathcal{L}(x^*, y^*, \mu^*)}{\partial x} = \frac{\partial f(x^*)}{\partial x} - \sum_i \lambda_i^* \frac{\partial h_i(x^*)}{\partial x} - \sum_j \mu_j^* \frac{\partial g_j(x^*)}{\partial x} = 0$$

1st order condition or stationarity

$$\left.\begin{aligned} h_i(x^*) - c_i &= 0 \\ g_j(x^*) - d_j &\geq 0 \end{aligned}\right\}$$

Primal feasibility conditions

$$\mu_j^* \geq 0$$

Dual feasibility conditions

$$\mu_j^*(g_j(x^*) - d_j) = 0$$

Complementary slackness conditions

$$\forall i, j.$$

On the meaning of KKT conditions

$$\mu_j^* \geq 0$$ Dual feasibility conditions

$$\mu_j^*(g_j(x^*) - d_j) = 0$$ Complementary slackness conditions

Only two options are possible:

(a) The constraint is **not binding**. It has no effect on the solution. We can remove the constraint

$$\mu_j^* = 0 \qquad \frac{\partial f(x^*)}{\partial x} = 0$$

(b) The constraint is **binding**. It modifies the solution. We cannot improve the objective along the constraint

$$\mu_j^* \neq 0 \qquad g_j(x^*) - d_j = 0 \qquad \mu_j^* > 0 \qquad \frac{\partial f(x^*)}{\partial x} = \mu_j^* \frac{\partial g(x^*)}{\partial x}.$$

Economic dispatch (or one-node dispatch optimisation)

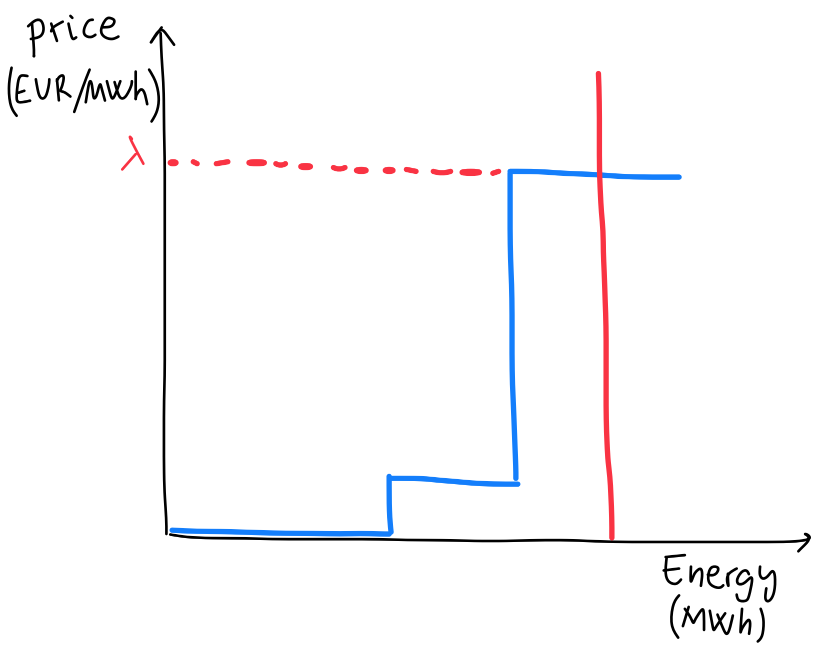
Types of optimisation problems and course structure

<table>
<tr><th></th><th>One node</th><th colspan="4">Network</th></tr>
<tr><td rowspan="2">One time step</td><td rowspan="2">Economic dispatch or
One-node dispatch optimisation
(Lecture 2)</td><td colspan="2">Power</td><td rowspan="2">Gas flow
(Lecture 7)</td><td rowspan="2">Heat flow
(Lecture 9)</td></tr>
<tr><td>Linearised
AC power flow
(Lecture 5)</td><td>AC power flow
(Lecture 6)</td></tr>
<tr><td>Multiple time steps</td><td>Multi-period optimisation

Join capacity and dispatch optimisation in one node
(Lecture 3)</td><td colspan="4">Join capacity and dispatch optimisation in a network
(Lecture 10)</td></tr>
</table>

Economic dispatch example

	Wind	Solar	Gas
Variable cost o_s (EUR/MWh)	0	5	50
Installed Capacity G_s (MW)	2	1	1
CF_s	0.5	0.5	1
Calculate generation g_s (assuming demand d = 1.5 MWh)	1	0.5	0
Calculate generation g_s (assuming demand d = 2 MWh)	1	0.5	0.5



Economic dispatch is also called merit-order dispatch (because we rank the generators based on their merit)

Economic dispatch is used to decide which generators produce energy in a market. It assumes ideal lossless ("copperplate") network within the area belonging to the market. After the economic dispatch, the system operators run AC power flow (including N-1 security constraint). If they identify anything that is physically impossible, redispatch takes place (e.g. via ancillary markets)

Economic dispatch or one-node dispatch optimisation (I)

Assume we have a set of generators s (e.g., onshore wind, solar PV, gas power plant…) each of them has an installed capacity G_s and a linear variable cost o_S. The economic dispatch consists in calculating the optimal dispatch (how much energy is being produced by each generator g_s) to supply the demand d in a certain hour while minimising the total system cost.

$$\min_{g_s} \sum_s o_s g_s$$

subject to:

$$\sum_s g_s - d = 0 \leftrightarrow \lambda$$

$$g_s \geq 0 \leftrightarrow \underline{\mu_s}$$

$$-g_s + G_s \geq 0 \leftrightarrow \overline{\mu_s}.$$

For renewable generators, the installed capacity is multiplied by the capacity factor: $-g_s + CF_s G_s \leq 0 \leftrightarrow \overline{\mu_s}.$

Economic dispatch or one-node dispatch optimisation (II)

Assume we have a set of generators s (e.g., onshore wind, solar PV, gas power plant…) each of them has an installed capacity G_s and a linear variable cost o_S. The economic dispatch consists in calculating the optimal dispatch (how much energy is being produced by each generator g_s) to supply the demand d in a certain hour while minimising the total system cost.

$$\min_{g_s} \sum_s o_s g_s$$

subject to:

$$\sum_s g_s - d = 0 \leftrightarrow \lambda$$

$$-g_s + G_s \geq 0 \leftrightarrow \overline{\mu_s}.$$

To find a solution, we start by building the Lagrangian function

$$\mathcal{L}(x, \lambda, \mu) = \sum_s o_s g_s - \lambda(\sum_s g_s - d) - \sum_s \overline{\mu_s}(-g_s + G_s)$$

We derive the Lagrangian and make the derivative equal to zero

$$\frac{\partial \mathcal{L}}{\partial g_s} = o_s - \lambda^* + \overline{\mu_s^*} = 0.$$

The inequality constraint can be binding ($\overline{\mu_s^*} > 0$) when the installed capacity is limiting the generation or not-binding ($\overline{\mu_s^*} = 0$).

The most expensive generator s whose capacity is not binding sets the price because for that generator $s1$ $\overline{\mu_{s1}^*} = 0$ $\lambda^* = o_{s1}$

Economic dispatch or one-node dispatch optimisation (III)

$$
\left\{
\begin{aligned}
&\min_{g_s} \sum_s o_s g_s \\
&\text{subject to:} \\
&\sum_s g_s - d = 0 \leftrightarrow \lambda \\
&g_s \geq 0 \leftrightarrow \underline{\mu_s} \\
&-g_s + G_s \geq 0 \leftrightarrow \overline{\mu_s}.
\end{aligned}
\right.
$$

λ represents the change in the objective function at the optimal solution with respect to a small change in the constraint.

Small change in constraint:

$$d^* = d^* + 1 \text{ MWh}$$

Change in objective function:

$$\text{System cost}^* = \text{System cost} + \Delta\text{System cost}$$

λ represents the cost of 1 MWh, i.e. the electricity price

So far, this set of equations excludes the consideration of any network constraints (e.g. line limits), and any additional security constraints, and additional generation constraints (CO_2 emissions, ramp limits …)

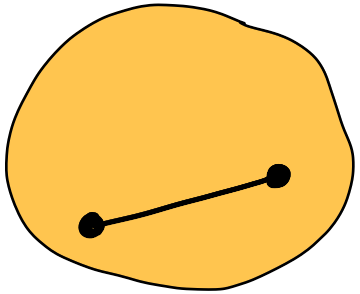
See Problems 2.2+2.3

Convex optimisation problems

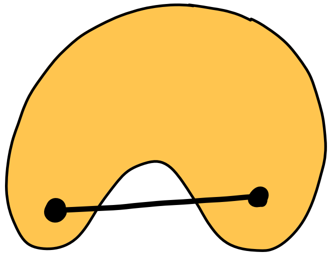
Convex optimisation problem

Convex set : A convex set X contains every line segment between two points in the set, i.e.
for all $x_1, x_2 \in \mathbb{R}^N$ and $\alpha \in [0,1]$: $\alpha x_1 + (1-\alpha)x_2 \in X$.

convex



non-convex



$$
\begin{cases}
\min_x f(x) \\
\text{subject to:} \\
h_i(x) - c_i = 0 \leftrightarrow \lambda_i \\
g_j(x) - d_j \geq 0 \leftrightarrow \mu_j .
\end{cases}
$$

An optimisation problem is convex if:

- The objective function $f(x)$ is convex
- The equality constraints $h_i(x)$ are affine (i.e. composed of a linear function and a constant)
- The inequality constrains $g_j(x)$ are convex

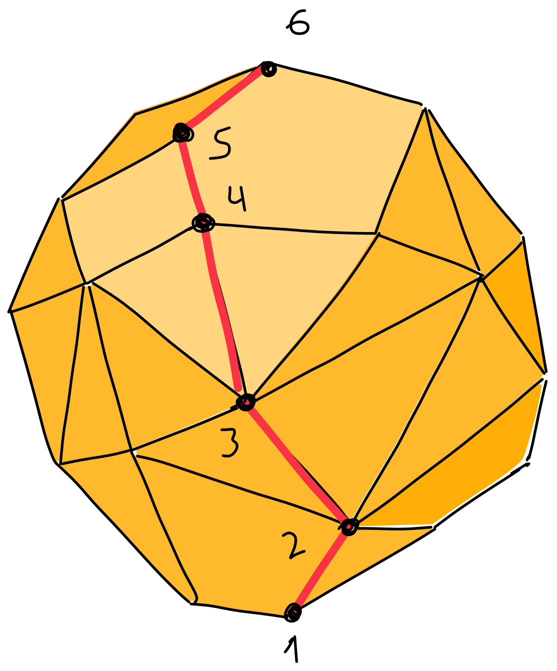
Convex optimisation problems have **one single optimum which is the global optimum.**

Optimisation algorithms and computational challenges (I)

If all the constraints are affine, the feasible space is a multi-dimensional polyhedron with only flat sides.

The optimum always occurs at one of the corners or vertices.

The simplex algorithm operates by descending the vertices of the polyhedron.

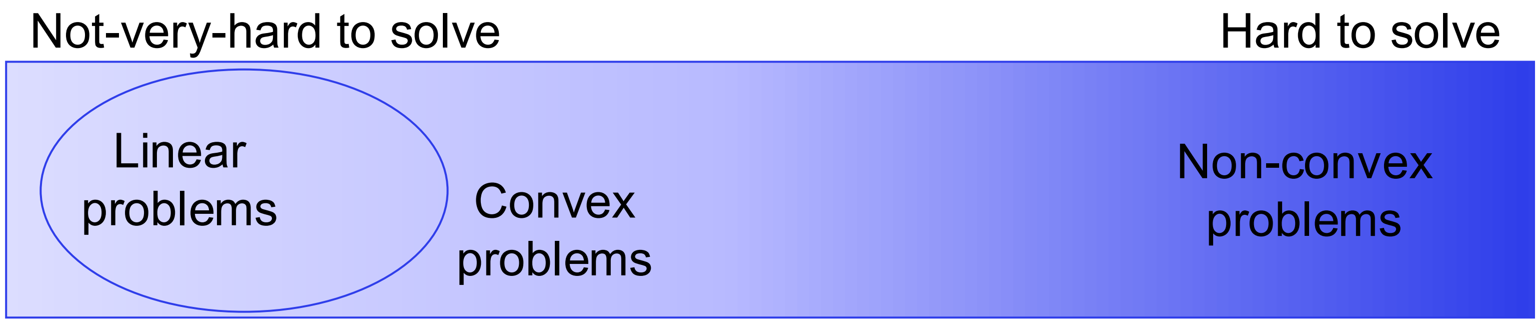


Optimisation algorithms and computational challenges (II)

Convex optimisation problems can be efficiently solved because it is not necessary to check numerous local minima to find the global minimum.

Usually convex problems admit polynomial-time algorithms.

Polynomial-time algorithms require a number of operations that is polynomial in both the number of variables and the constraints.

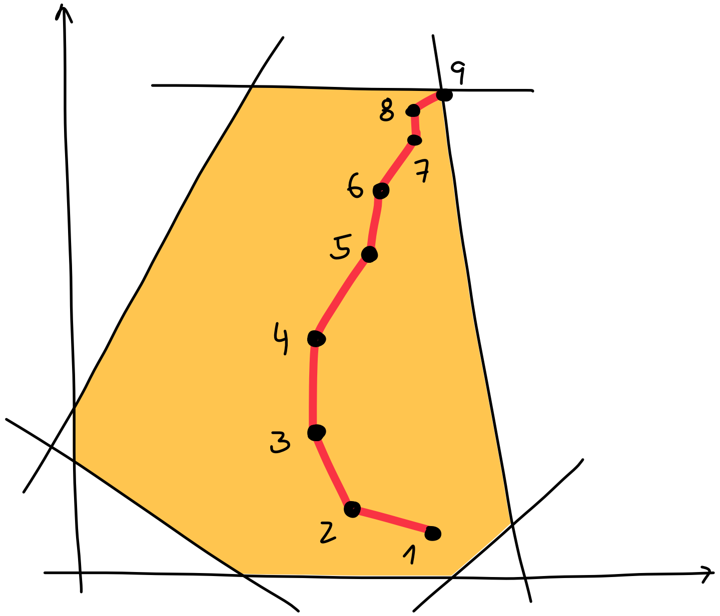


Barrier interior-point methods

We can use interior-point methods which converge faster.

The objective is to iteratively approach the optimal solution from the interior of the feasible space.

A barrier term is added to the objective function that penalises solutions that come close to the boundary.

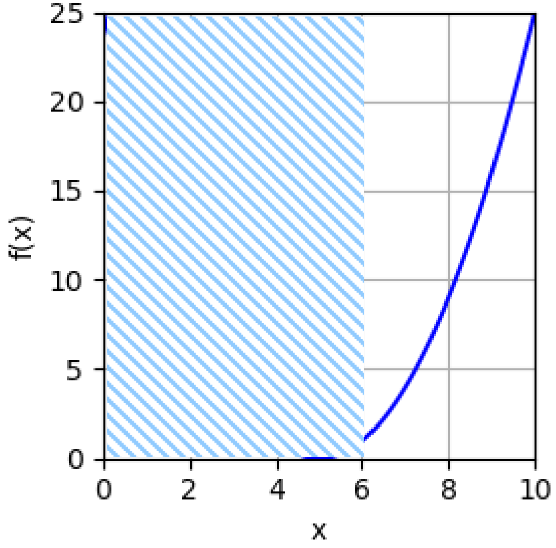


Karmarkar's interior-point method can run in polynomial time for linear problems.

Barrier interior-point methods: Example

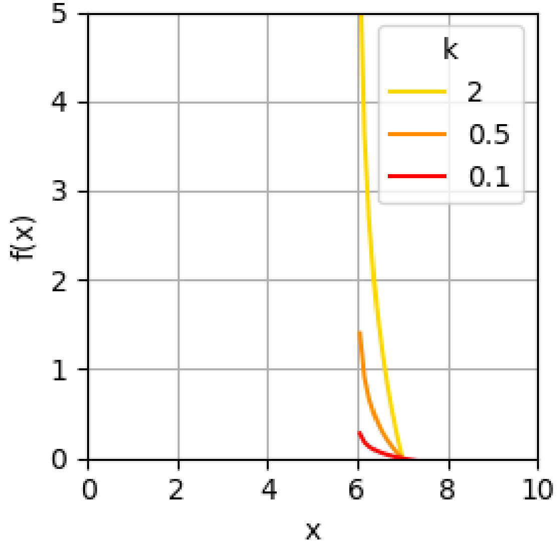
Let's recover our initial optimisation problem

$$\begin{aligned} &\min f(x) = (x-5)^2 \\ &\text{subject to:} \\ &\qquad x \geq 6 \end{aligned}$$



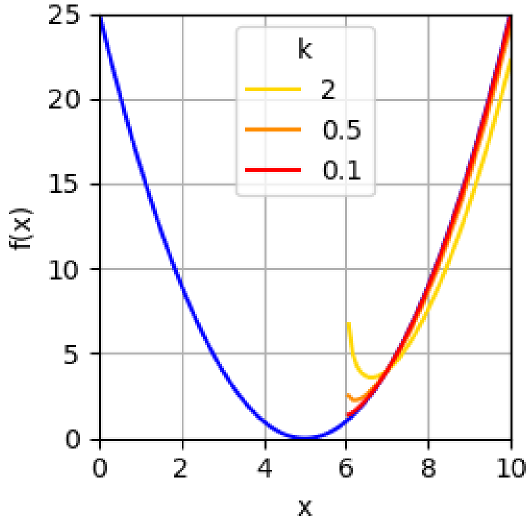
We can add a barrier term that increases when the solution approaches the boundary

$$\text{penalty} = -k \cdot \log(x-6)$$



Now, we can substitute the constraint by the barrier term

$$f(x) = (x-5) - \underbrace{k \cdot \log(x-6)}_{\text{barrier term}}$$

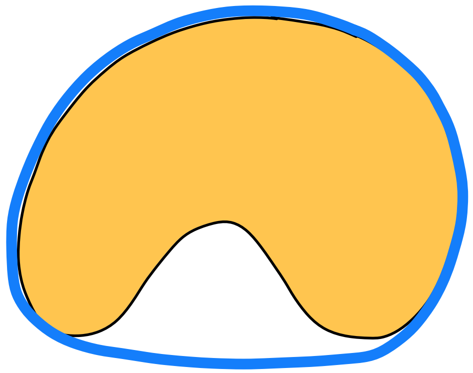


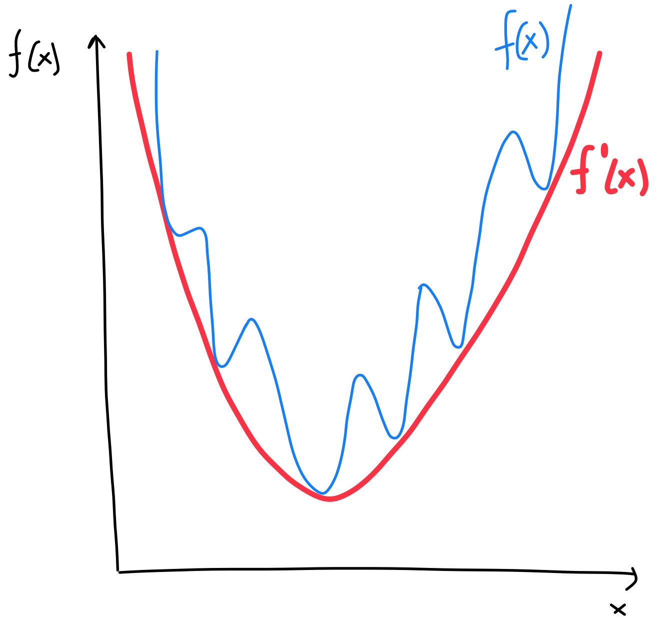
By reducing the penalty, the algorithm converges to the optimal point at the boundary

Convex relaxation

To solve a non-convex optimisation problem, we typically create a convex relaxation problem, and we say that we “convexify” the problem.

Convex relaxation: we create an alternative problem by relaxing some of the constraints.





Additional materials: Short videos

Visually Explained: Convexity and The Principle of Duality
https://www.youtube.com/watch?v=d0CF3d5aEGc&list=PLP3dxscCx69JuM3Ppv9CD49IN1Z_a2Avq&index=9

Visually Explained: The Karush–Kuhn–Tucker (KKT) Conditions and the Interior Point Method for Convex optimisation https://www.youtube.com/watch?v=uh1Dk68cfWs&t=1169s

Understanding Lagrange Multipliers Visually

https://www.youtube.com/watch?v=5A39Ht9Wcu0

Problems for this lecture

Complete the multiple-choice quiz in Week 2 in DTU Learn.

Start with the introduction to linopy:
https://aleks-g.github.io/integrated-energy-grids/intro-linopy.html

To be presented next week:

Problems 2.1 and 2.2 (**Group 4**)
Problems 2.3 (**Group 5**)